

Министерство образования и науки РФ
ФГБОУ ВПО «Удмуртский государственный университет»
Факультет информационных технологий
и вычислительной техники

С.Ю. Купчинаус

**ФОРМИРОВАНИЕ
КОНСТРУКТИВНО-ЛОГИЧЕСКОЙ КОМПЕТЕНТНОСТИ
БУДУЩИХ СПЕЦИАЛИСТОВ
ИНФОРМАТИКИ И УПРАВЛЕНИЯ**

Монография



Ижевск
2011

ББК 73в + 74.580.2
УДК 002.6–027.21 + 14.35 + 378.2
К927

*Рекомендовано к изданию Редакционно-издательским советом
УдГУ*

Рецензент – д.ф.-м.н., профессор, заведующий кафедрой
теоретических основ информатики ФИТиВТ УдГУ
А.П. Бельтюков

Купчинаус С.Ю.

К927 Формирование конструктивно-логической компетентности
будущих специалистов информатики и управления:
монография. Ижевск: Изд-во «Удмуртский университет», 2011.
– 152 с.

Монография посвящена изучению одного из видов
практического мышления – конструктивно-логического и
связанных с ним аспектов выявления, формирования и развития
соответствующих компетенций. Издание адресовано
преподавателям информатики и основ управления в различных
областях, поддерживающим развивающую парадигму обучения,
а также и студентам старших курсов направлений информатики,
нацеленным на изучение и инновационное развитие различных
видов управления – от педагогического и организационного до
социотехнического и социального.

ISBN 978-5-4312-0077-9

УДК 002.6–027.21 + 14.35 + 378.2
ББК 73в + 74.580.2

© С.Ю. Купчинаус, 2011
© ФГБОУ ВПО «Удмуртский
государственный университет», 2011

Содержание

Введение	5
Глава 1. Практическое мышление и его разновидности. Моделирование процесса развития конструктивно-логического мышления у студента при обучении программированию	8
1.1. Подход к моделированию процесса развития конструктивно-логического мышления у студентов.	21
1.2. Модель процесса развития конструктивно-логического мышления у студентов при изучении программирования ...	36
1.3. Программа развития конструктивно-логического мышления у студентов при изучении основ программирования	45
Выводы	49
Глава 2. Опытнo-экспериментальная работа по обучению программированию и развитию конструктивно-логического мышления студентов	51
2.1. Организация и методика констатирующего этапа экспериментального исследования.	52
2.2. Методическое обеспечение и организация формирующего эксперимента. Анализ и интерпретация результатов формирующего эксперимента.	69
2.3. Контролирующий этап - проверка эффективности дидактических условий развития конструктивно-логического мышления	72
2.4. Методические рекомендации по обучению будущих педагогов-преподавателей математики и информатики	78
Выводы	86

Глава 3. Элементы дидактической системы формирования конструктивно-логической компетентности: от выявления одаренности до начальных уровней профессионализма в информатике	90
3.1. Мышление информатика-программиста и управленческое мышление как виды практического мышления. Информационное моделирование и развитие управленческого мышления	90
3.2. Проектирование содержания при обучении программированию. Микроанализ программирования как последовательности проблемных ситуаций управления	93
3.3. Анализ информационного моделирования как вида организационного управления. Управление и информационное моделирование – два вида конструктивно-логической деятельности над предметной областью	102
3.4. Рефлексивный анализ управления и программирования. Подход к обучению студентов в рамках парадигмы персональной поддержки управленческой деятельности	110
3.5. Три этапа и три парадигмы компьютеризации системы управления вуза (пример реализации предлагаемого подхода) ..	113
3.6. Компетентности и компетентностный подход в подготовке информатиков и специалистов управления	116
Выводы	123
Заключение	125
Список использованной литературы	127
Приложения	139
Приложение 1. Модель процесса развития конструктивно-логического мышления у студента при изучении программирования	140
Приложение 2. Программа экспериментального обучения студентов программированию	142
Приложение 3 . Специальный тест по оценке компетенций в алгоритмизации и программировании	148

Введение.

Конструктивно-логическая деятельность и подготовка к ней

Познание окружающей действительности и управление ею в интересах общественного прогресса и развития личности требуют от любого человека определенного набора знаний, умений и навыков. Сегодня основным содержанием профессиональной деятельности работников в любой сфере деятельности является управление некоторой предметной областью через анализ и моделирование ее процессов и явлений конструктивно-логическими средствами. Для этого необходим определенный уровень конструктивно-логического мышления (КЛМ). Именно такое мышление требуется учителю и ученому, инженеру и экономисту, рядовому менеджеру и руководителю организации при анализе, описании и оптимизации управляемой ситуации в предметной области (ПрО).

Конструктивно-логическое мышление требуется и для эффективной работы с современными программными системами, поддерживающими деятельность широкого круга специалистов. Сегодня рабочее место специалиста оснащено персональным компьютером (ПК), который по сети объединен с другими ПК, таким образом, автоматизированные рабочие места (АРМ) объединены в единое целое, в единую среду для решения задач управления различного уровня. Содержанием работы специалистов на АРМ является сбор данных, анализ, описание и измерение параметров управляемой ситуации, ее структурирование на составляющие, конструирование различных вариантов управления и их оценка – все вместе это моделирование ситуации с целью поиска вариантов решения задачи управления и выбор наиболее оптимального варианта принятия решения (ПР).

Одним из наиболее эффективных средств моделирования является программное моделирование или программирование задачи для ЭВМ. В рамках доминирующей так называемой алгоритмической парадигмы исследуемый процесс или явление должны быть представлены в виде алгоритма, воплощенного в компьютерную программу. Отсюда, алгоритмический стиль мышления (АСМ) может рассматриваться как необходимая составляющая часть конструктивно-логического мышления в случае компьютерной поддержки деятельности современного специалиста.

В то же время нет необходимости рассматривать всех специалистов, занятых программным моделированием и управлением

в различных областях практики, как программистов - сегодня для широкого круга специалистов создаются и совершенствуются специальные программные системы - удобные средства описания и анализа процессов и объектов их предметной области.

Актуальность исследования. Конструктивный характер деятельности современного человека есть потребность времени. Конструктивность и логичность это - единственный способ решать профессиональные задачи в любой сфере современной жизни. Независимо от сферы деятельности современному специалисту требуется всестороннее развитие такого вида практического мышления как конструктивно-логическое.

При переходе к информационному обществу в условиях постоянного взаимодействия с компьютерами алгоритмический стиль мышления становится необходимой основой действий современного человека. Компьютеры и микропроцессоры – алгоритмические устройства заставляют человека-оператора действовать по строгим однозначным правилам – алгоритмам, точнее, предписаниям алгоритмического типа. В рамках данного исследования два понятия – конструктивно-логическое мышление и алгоритмический стиль мышления объединены в комплекс «конструктивно-логическое мышление - алгоритмический стиль мышления» (КЛМ-АСМ).

Влияние компьютерной техники и информационных технологий на мышление общепризнанно и логично предположить, что это влияние может быть целенаправленным, организованным, а обучение алгоритмизации и программированию могут рассматриваться как эффективное средство развития этого комплекса. Наблюдения показывают, что уроки по предмету «Основы информатики и ВТ» в средней школе и занятия по информатике и программированию в вузе, как правило, не имеют выраженной *развивающей направленности*, а чаще всего сводятся к освоению технических умений и навыков работы с компьютером или программированию типовых задач по образцу.

Таким образом, существуют *противоречия* между:

- 1) потребностью в специалистах современного типа, способных решать профессиональные задачи педагогического или социального управления, и недостаточным уровнем подготовленности выпускников вузов к работе в быстроменяющихся условиях рыночной экономики;
- 2) необходимостью развития конструктивно-логического мышления у студентов педагогических специальностей и

недостаточной разработанностью адекватных для этого дидактических условий.

Выявленные противоречия позволили определить **проблему исследования:** каковы дидактические условия развития конструктивно-логического мышления у студентов различных специальностей, изучающих информатику и информационное моделирование применительно к своей профессиональной области, и, в частности, основы алгоритмизации и программирования?

Цель настоящего исследования – выявить, теоретически обосновать дидактические условия, способствующие формированию и развитию конструктивно-логических компонентов профессиональной компетентности, в том числе, конструктивно-логического мышления (КЛМ) и алгоритмического стиля мышления (АСМ) у студентов вуза, и экспериментально проверить эффективность их реализации.

Заложить основы дидактической системы формирования конструктивно-логических компетенций будущих специалистов в области информатики и управления.

Объект исследования: процесс развития конструктивно-логического мышления студентов-педагогов и формирования названных компетенций.

Предметом исследования являются дидактические условия, способствующие развитию конструктивно-логического мышления у студентов при изучении основ программирования и информационного моделирования, и элементы соответствующей дидактической системы.

Проблема состоит в недостаточном уровне развития навыков конструктивно-логического мышления студентов и выпускников вуза, что предопределяет недостаточную эффективность работы специалиста, занятого управлением и моделированием в определенной предметной области.

Способность мыслить структурировано, точно и формально, если это нужно, становится одним из важных признаков общей культуры человека в современном высокотехнологизированном мире. Эта способность приобретает весомое значение при освоении современных профессий, связанных с операторской деятельностью, которой присуще оперативное мышление.

Компетентности и компетентностный подход в современном образовании [132, 144].

Компетентностный подход впервые начал разрабатываться в Англии. Это был подход, который порождался и осмысливался не внутри образования, а был ответом на конкретный заказ профессиональной сферы. В США в сфере бизнеса в 70-х годах прошлого века стали использоваться понятия «компетенция» и «ключевые компетенции» в связи с проблемой определения качеств успешного профессионала.

Изначально компетенции стали противопоставляться специальным профессиональным знаниям и умениям. То есть начали рассматриваться как самостоятельные универсальные составляющие *любой* успешной профессиональной деятельности. Естественно, возник вопрос: можно ли научить компетенциям? Таким образом, проблематика компетенций попала в образование и со временем заняла в нем ведущее место.

Сфера образования, начиная с Я.А.Коменского, работала с основными единицами – знаниями, умениями и навыками (ЗУН). Профессиональная сфера работала с другими единицами – компетенциями (**профессиональные компетенции, ПК**). В этом смысле профессия дает ответ, какой компетентности должен быть человек или какова сфера его компетенции. Поэтому профессиональная сфера оперирует компетенциями, а образование – знаниями, умениями и навыками. И когда профессиональная сфера может точно на уровне заказа однозначно формулировать свои претензии к образованию, то задача образования заключается в том, как перекомпоновать знания, умения и навыки в определенные компетенции, которые требуются в профессиональной сфере.

Что такое компетентность? Устоявшегося определения для содержания понятия «компетенция» или «ключевая компетенция» до сих пор нет. Не существует и единой, принятой всеми классификации компетенций. Тем не менее, большинство авторов связывают компетентность с **эффективным выполнением какой-либо деятельности или действия.**

В Глоссарии терминов Европейского фонда образования (ЕФО, 1997) компетенция определяется как:

1. Способность делать что-либо хорошо или эффективно.
2. Соответствие требованиям, предъявляемым при устройстве на работу.

3. Способность выполнять особые трудовые функции.

То есть компетентность - это личностное свойство, характеристика, даваемая человеку в результате оценки эффективности/результативности его действий, направленных на разрешение определенного круга значимых для данного сообщества задач/проблем. Знания, навыки, способности, мотивы, ценности и убеждения рассматриваются как возможные составляющие компетентности, но сами по себе еще не делают человека компетентным.

В этом определении усматривается два подхода к содержанию понятия «компетенция». **Одни** исследователи делают **акцент** на компетенции как интегральном *личностном качестве человека* (характеристика человека), **другие** - на описании *составляющих его деятельности*, ее различных аспектов, которые и позволяют ему успешно справиться с решением проблем.

А.В. Хуторской [131] дает следующее определение понятия «компетенция»

Компетенция – отчужденное, заранее заданное социальное требование (норма) к образовательной подготовке ученика, необходимой для его эффективной продуктивной деятельности в определенной сфере.

Компетентность – владение, обладание учеником соответствующей компетенцией, включающее его личностное отношение к ней и предмету деятельности. Компетентность – уже состоявшееся качество личности (совокупность качеств) ученика и минимальный опыт деятельности в заданной сфере.

Таким образом, **компетентностный подход** предполагает не усвоение учеником отдельных друг от друга знаний и умений, а овладение ими в комплексе. В связи с этим меняется, точнее, по иному определяется система методов обучения. В основе отбора и конструирования методов обучения лежит структура соответствующих компетенций и функций, которые они выполняют в образовании.

Компетенции профессиональные следует отличать от образовательных компетенций, т.е. от тех, которые моделируют деятельность ученика для его полноценной жизни в будущем.

Образовательная компетенция – требование к образовательной подготовке, выраженное совокупностью взаимосвязанных смысловых ориентаций, знаний, умений, навыков и опыта деятельности ученика по отношению к определенному кругу объектов реальной

действительности, необходимых для осуществления личностно и социально значимой продуктивной деятельности.

Компетенции для ученика – это образ его будущего, ориентир для освоения. Но в период обучения у него формируются те или иные составляющие этих «взрослых» компетенций, и чтобы не только готовиться к будущему, но и жить в настоящем, он осваивает эти компетенции с образовательной точки зрения. Образовательные компетенции относятся не ко всем видам деятельности, в которых участвует человек, например, взрослый специалист, а только к тем, которые включены в состав общеобразовательных областей и учебных предметов. Такие компетенции отражают предметно-деятельностную составляющую общего образования и призваны обеспечивать комплексное достижение его целей.

Ключевые компетенции

Сам термин «**ключевые компетенции**» указывает на то, что они являются «ключом», основанием для других, более конкретных и предметно ориентированных. Предполагается, что ключевые компетенции носят *надпрофессиональный и надпредметный характер* и необходимы в любой области деятельности.

В Оксфордско-Кембриджской образовательной программе «Ключевые компетенции 2000» указывается, что *ключевые компетенции* используются в повседневной жизни при осуществлении деятельности в области образования, на рабочем месте или при получении профессиональной подготовки.

В европейском проекте «Определение и отбор ключевых компетенций» (DeSeCo) **ключевые компетенции** определяются как важные **«во многих жизненных сферах и служащие залогом жизненного успеха и эффективного функционирования общества»**.

Стратегия модернизации образования в РФ в основу обновленного содержания общего образования положила «ключевые компетентности»: «Основным результатом деятельности образовательного учреждения должна стать не система знаний, умений и навыков *сама по себе*, а набор *заявленных государством* ключевых компетенций в интеллектуальной, общественно-политической, коммуникационной, информационной и прочих сферах».

Таким образом, *существенными признаками* ключевых компетенций являются:

1. Ключевые компетенции представляют собой различные универсальные ментальные средства, инструменты (способы, методы, приемы) достижения человеком значимых для него целей (результатов).

2. Ключевыми компетенциями в той или иной степени должен овладеть каждый член общества.

3. Ключевые компетенции позволяют человеку достигать результатов в не определенных, проблемных ситуациях. Они позволяют *самостоятельно* и в *сотрудничестве* с другими решать *проблемы*, то есть справляться с ситуациями, для разрешения которых *никогда нет полного комплекта наработанных средств*.

4. Определение и отбор ключевых компетенций осуществляется основными потребителями образовательных результатов на основе социологических исследований и общественного обсуждения и зависит от того, какие способности и качества человека являются ценными в данное время в данном обществе.

5. В современном западном обществе нормативную основу для отбора ключевых компетенций составляют базовые принципы прав человека, демократические ценности и цели, связанные с устойчивым развитием.

6. Компетенции проявляются и приобретаются человеком в деятельности, имеющей для него ценность.

Перечень ключевых компетенций основывается на главных целях общего образования, структурном представлении социального опыта и опыта личности, а также основных видах деятельности ученика, позволяющих ему овладевать социальным опытом, получать навыки жизни и практической деятельности в современном обществе.

Ценностно-смысловые компетенции (ЦСК). Это компетенции, связанные с ценностными ориентирами ученика, его способностью видеть и понимать окружающий мир, ориентироваться в нем, осознавать свою роль и предназначение, уметь выбирать целевые и смысловые установки для своих действий и поступков, принимать решения. Данные компетенции обеспечивают механизм самоопределения ученика в ситуациях учебной и иной деятельности. От них зависит индивидуальная образовательная траектория ученика и программа его жизнедеятельности в целом.

Ценностно-смысловые компетенции предполагают умения:

- формулировать собственные ценностные ориентиры по отношению к изучаемым учебным предметам и сферам деятельности; владеть способами самоопределения в ситуациях выбора на основе собственных позиций; уметь принимать решения, брать на себя ответственность за их последствия, осуществлять действия и поступки на основе выбранных целевых и смысловых установок;
- осуществлять индивидуальную образовательную траекторию с учетом общих требований и норм.

Общекультурные компетенции (ОК). Познание и опыт деятельности в области национальной и общечеловеческой культуры; духовно-нравственные основы жизни человека и человечества, отдельных народов; культурологические основы семейных, социальных, общественных явлений и традиций; роль науки и религии в жизни человека; компетенции в бытовой и культурно-досуговой сфере, например, владение эффективными способами организации свободного времени. Сюда же относится опыт освоения учеником картины мира, расширяющейся до культурологического и всечеловеческого понимания мира

Учебно-познавательные компетенции (УПК). Это совокупность компетенций ученика в сфере самостоятельной познавательной деятельности, включающей элементы логической, методологической, общеучебной деятельности. Сюда входят способы организации целеполагания, планирования, анализа, рефлексии, самооценки. По отношению к изучаемым объектам ученик овладевает креативными навыками: добытием знаний непосредственно из окружающей действительности, владением приемами учебно-познавательных проблем, действий в нестандартных ситуациях.

Учебно-познавательные компетенции:

- ставить цель и организовывать её достижение, уметь пояснить свою цель;
- организовывать планирование, анализ, рефлексия, самооценку своей учебно-познавательной деятельности;
- задавать вопросы к наблюдаемым фактам, отыскивать причины явлений, обозначать свое понимание или непонимание по отношению к изучаемой проблеме;
- ставить познавательные задачи и выдвигать гипотезы; описывать результаты, формулировать выводы;

- выступать устно и письменно о результатах своего исследования с использованием компьютерных средств и технологий (текстовые и графические редакторы, презентации);
- иметь опыт восприятия картины мира.

Информационные компетенции (ИК). Навыки деятельности по отношению к информации в учебных предметах и образовательных областях, а также в окружающем мире. Владение современными средствами информации (телевизор, магнитофон, телефон, факс, компьютер, принтер, модем, копир и т.п.) и информационными технологиями (аудио- видеозапись, электронная почта, СМИ, Интернет). Поиск, анализ и отбор необходимой информации, ее преобразование, сохранение и передача.

Коммуникативные компетенции (КК). Знание языков, способов взаимодействия с окружающими и удаленными событиями и людьми; навыки работы в группе, коллективе, владение различными социальными ролями. Ученик должен уметь представить себя, написать письмо, анкету, заявление, задать вопрос, вести дискуссию и др. Для освоения этих компетенций в учебном процессе фиксируется необходимое и достаточное количество реальных объектов коммуникации и способов работы с ними для ученика каждой ступени обучения в рамках каждого изучаемого предмета или образовательной области.

Коммуникативные компетенции:

- уметь представить себя устно и письменно, написать анкету, заявление, резюме, письмо, поздравление;
- уметь представлять свой класс, школу, страну в ситуациях межкультурного общения, в режиме диалога культур, использовать для этого знание иностранного языка;
- владеть способами взаимодействия с окружающими и удаленными людьми и событиями; выступать с устным сообщением, уметь задать вопрос, корректно вести учебный диалог;
- владеть разными видами речевой деятельности (монолог, диалог, чтение, письмо), лингвистической и языковой компетенциями;
- владеть способами совместной деятельности в группе, приемами действий в ситуациях общения; умениями искать и находить компромиссы;
- иметь позитивные навыки общения в поликультурном, полиэтническом и многоконфессиональном обществе, основанные на

знании исторических корней и традиций различных национальных общностей и социальных групп.

Социально-трудовые компетенции (СТК). Выполнение роли гражданина, наблюдателя, избирателя, представителя, потребителя, покупателя, клиента, производителя, члена семьи. Права и обязанности в вопросах экономики и права, в области профессионального самоопределения. В данные компетенции входят, например, умения анализировать ситуацию на рынке труда, действовать в соответствии с личной и общественной выгодой, владеть этикой трудовых и гражданских взаимоотношений.

Компетенции личностного самосовершенствования (КЛС) направлены на освоение способов физического, духовного и интеллектуального саморазвития, эмоциональной саморегуляции и самоподдержки. Ученик овладевает способами деятельности в собственных интересах и возможностях, что выражается в его непрерывном самопознании, развитии необходимых современному человеку личностных качеств, формировании психологической грамотности, культуры мышления и поведения.

Использование компетентностной модели в образовании предполагает принципиальные изменения в организации учебного процесса, в управлении им, в деятельности учителей и преподавателей, в способах оценивания образовательных результатов учащихся по сравнению с учебным процессом, основанным на концепции «усвоения знаний».

Дело в том, что целый ряд школьных умений и знаний уже не принадлежит никакому профессиональному занятию. Примером такого экзотического вида школьных занятий может быть целый предмет черчение. Здесь остро необходима ревизия содержания образования. Во многих странах традиционные курсы трудового обучения и домоводства были заменены курсами «Технология и дизайн», «Предпринимательство». И это *все* - часть того обновления школы, которое проходит под лозунгами компетентностного подхода.

В *компетентностном подходе* перечень необходимых компетенций определяется в соответствии с запросами работодателей, требованиями со стороны академического сообщества и широкого общественного обсуждения на основе серьезных социологических исследований. Овладение различного рода компетенциями становится основной целью и результатами процесса обучения.

Основной ценностью становится не усвоение суммы сведений, а освоение учащимися таких умений, которые позволяли бы им определять свои цели, принимать решения и действовать в типичных и нестандартных ситуациях.

Позиции субъектов при компетентностном подходе

Позиция учителя Принципиально изменяется и позиция учителя, преподавателя. Он перестает быть вместе с учебником носителем «объективного знания», которое он пытается передать ученику. Его главной задачей становится мотивировать учащихся на проявление инициативы и самостоятельности. Он должен организовать самостоятельную деятельность учащихся, в которой каждый мог бы реализовать свои способности и интересы. Фактически он создает условия, развивающую среду, в которой становится возможным выработка каждым учащимся на уровне развития его интеллектуальных и прочих способностей определенных компетенций.

Развитие ребенка

Меняется и смысл термина «развитие». Индивидуальное развитие каждого человека связано в первую очередь с приобретением умений, к которым у него уже есть предрасположенность (способность), а не с приобретением тематической информации, которая не только никогда не понадобится в практической жизни, но и, по сути, не имеет никакого отношения к его индивидуальности.

Оценка уровня достижений ключевых компетенций

В целом освоенность тех или иных компетенций в учебном процессе можно оценивать как по результату разрешения проблемных ситуаций (в соответствии с заранее разработанными критериями успешности достижения этих результатов), так и в самом процессе их применения, использования, опять же по известным критериям.

Оценивают уровень владения конкретными компетенциями в таких процедурах, как *наблюдение за исполнением действий учащегося* в конкретных ситуациях, связанных с формированием, определенных компетенций (исследование, дискуссия, выступление и т.д.), письменного экзамена - написание эссе, а также и в других видах письменных экзаменов, решение тестовых заданий типа PISA.

Таким образом, компетентностный подход является усилением прикладного, практического характера образования всех уровней и форм (в том числе и предметного обучения). Это направление возникло из простых вопросов о том, какими результатами образования может воспользоваться школьник вне школы, выпускник вуза – начиная работать в своей профессиональной области. Ключевая мысль этого направления состоит в том, что для обеспечения «отдаленного эффекта» образования все, что изучается, должно быть включено в процесс употребления, использования. Особенно это касается теоретических знаний; которые должны перестать быть мертвым багажом и стать практическим средством объяснения явлений и решения практических ситуаций и проблем.

В связи с лавинообразным развитием информационных технологий, когда профессия информатика становится массовой, так называемая «проблема детской одаренности в области информационных технологий» (Бабаева Ю.Д., Войскунский А.Е. [1]) все более привлекает внимание исследователей.

Ранее [47-70] нами было введены более корректные и более емкие понятия «конструктивно-логического мышления» (КЛМ) и «конструктивно-логическая деятельность» (КЛД), к которой отнесено информационное моделирование некоторой ПрО и, в частности, алгоритмизация и программирование задач для ЭВМ, а также управление некоторой предметной областью. Отсюда естественным образом родился связанный блок понятий и терминов - «конструктивно-логическая одаренность» (КЛО) у дошкольников, «конструктивно-логические способности» (КЛС) школьников и студентов, «конструктивно-логическая компетенция (КЛК) как профессиональная компетенция для различных областей деятельности, как ключевая компетенция в обучении информатика и программиста, специалистов для различных областей организационного и социотехнического управления. Естественной видится и последовательность изучения и построения этапов формирования КЛК: 1) выявление одаренности КЛО; 2) развитие КЛО до способностей КЛС (от начальной школы до выпуска из школы); 3) формирование КЛК студента до начального профессионального уровня.

Глава 1. Практическое мышление и его разновидности. Моделирование процесса развития конструктивно-логического мышления у студента при обучении программированию

Основываясь на концепции практического мышления Б.М.Теплова [118], рассмотрим *философско-психологический аспект* проблемы, который позволяет определить цели и способы, специфические для мыслительной деятельности конструктивно-логического типа.

В психологии вопросы мышления ставились обычно весьма абстрактно [127]. Происходило это потому, что при исследовании мышления имелись в виду лишь те задачи и те мыслительные операции, которые возникают при чисто интеллектуальной, теоретической деятельности. Большинство психологов - сознательно или бессознательно - принимали за единственный образец умственной работы работу ученого, философа, вообще теоретика. Между тем в работе любого организатора, администратора, производственника, хозяйственника и т. д. постоянно встают вопросы, требующие напряженной мыслительной деятельности. Исследование практического мышления представляет для психологии не меньшую важность и не меньший интерес, чем исследование мышления теоретического [127].

«Проблема практического интеллекта в психологии ставилась часто, но в другом плане. Говоря о практическом интеллекте, подразумевали некий совсем особый интеллект, работающий иными механизмами, чем те, которыми пользуется обычное теоретическое мышление. Проблема практического интеллекта сужалась до вопроса о так называемом наглядно-действенном, или сенсомоторном, мышлении. Под этим разумелось мышление, которое, во-первых, неотрывно от восприятия, оперирует лишь непосредственно воспринимаемыми вещами и теми связями вещей, которые даны в восприятии, и, во-вторых, неотрывно от прямого манипулирования с вещами, неотрывно от действия в моторном, физическом смысле этого слова. При таком мышлении человек решает задачу, глядя на вещи и оперируя с ними» [127].

Понятие наглядно-действенного мышления — очень важное понятие. Крупнейшим приобретением материалистической психологии является установление того факта, что и в филогенезе, и в онтогенезе генетически первой ступенью мышления может быть только наглядно-действенное мышление [127]. «Интеллектуальная

деятельность формируется сначала в плане действия; она опирается на восприятие и выражается в более или менее осмысленных, целенаправленных предметных действиях. Можно сказать, что у ребенка на этой ступени (имеются в виду первые годы жизни) лишь наглядно-действенное мышление, или сенсомоторный интеллект» (Рубинштейн С.Л., 1940, с. 315). Очевидно, однако, что понятие сенсомоторного интеллекта не имеет прямого отношения к вопросу об особенностях практического мышления. Человек, занятый организационной работой, решает стоящие перед ним задачи, опираясь вовсе не на непосредственное восприятие вещей и прямое манипулирование с ними. Объекты его умственной деятельности (взаимоотношения групп людей, занятых в каком-либо производстве, способы руководства этими группами и установления связи между ними и т. п.) таковы, что они едва ли поддаются непосредственному восприятию и не поддаются физическому, моторному оперированию с ними. Скорее можно предположить сенсомоторный интеллект у ученого-экспериментатора в области, например, физики или химии, чем у практика-администратора. Участие в мышлении восприятия и движения различно в разных конкретных видах деятельности, но степень этого участия никак не является признаком, отличающим практическое мышление от теоретического.

Отличие между этими двумя типами мышления нельзя искать в различиях самих механизмов мышления, в том, что тут действуют «два разных интеллекта». Интеллект у человека один, и едины основные механизмы мышления, но различны *формы* мыслительной деятельности, поскольку различны задачи, стоящие в том и другом случае перед умом человека. Именно в таком смысле можно и должно говорить в психологии о практическом и теоретическом уме [127].

Различие между теоретическим и практическим мышлением заключается в том, что они по-разному связаны с практикой; не в том, что одно из них имеет связь с практикой, а другое - нет, а в том, что характер этой связи различен. Работа практического мышления в основном направлена на разрешение частных, конкретных задач - организовать работу данного завода, разработать и осуществить план сражения и т. п., - тогда как работа теоретического мышления направлена в основном на нахождение общих закономерностей - принципов организации производства, тактических и стратегических закономерностей и т. п.

Работа теоретического ума сосредоточена преимущественно на

первой части целостного пути познания: на переходе от живого созерцания к абстрактному мышлению, на (временном!) отходе, отступлении от практики. Работа практического ума сосредоточена главным образом на второй части этого пути познания: на переходе от абстрактного мышления к практике, для которого и производится теоретический отход.

И теоретическое и практическое мышление связано с практикой, но во втором случае связь эта имеет более непосредственный характер. Работа практического ума непосредственно вплетена в практическую деятельность и подвергается ее непрерывному испытанию, тогда как работа теоретического ума обычно подвергается такой проверке лишь в конечных результатах. Отсюда та своеобразная *ответственность*, которая присуща практическому мышлению. Теоретический ум отвечает перед практикой лишь за конечный результат своей работы, тогда как практический ум несет ответственность в самом процессе мыслительной деятельности. Ученый-теоретик может выдвигать разного рода рабочие гипотезы, испытывать их, иногда в течение очень длительного срока, отбрасывать те, которые себя не оправдывают, заменять их другими и т. д. У практика возможности пользоваться гипотезами несравненно более ограничены, так как проверяться эти гипотезы должны не в специальных экспериментах, а в самой жизни, и — что особенно важно — практический работник далеко не всегда имеет время для такого рода проверок. Жесткие условия времени — одна из самых характерных особенностей работы практического ума [127].

Сказанного достаточно, чтобы поставить под сомнение очень распространенное убеждение в том, что наиболее высокие требования к уму предъявляют теоретические деятельности: наука, философия, искусство. Психологи начала XX в. наиболее высоким проявлением умственной деятельности считали, как правило, работу ученого. Во всех случаях теоретический ум рассматривался как высшая возможная форма проявления интеллекта. Практический же ум, даже на самых высоких его ступенях — ум политика, государственного деятеля, полководца, — рассматривался с этой точки зрения как более элементарная, легкая, как бы менее квалифицированная форма интеллектуальной деятельности [127].

«Это убеждение глубоко ошибочно» (Б.М.Теплов). Если различие между практическим и теоретическим умом понимать так, как об этом сказано выше, то нет ни малейшего основания считать работу

практического ума более простой и элементарной, чем работу ума теоретического. Да и фактически высшие проявления человеческого ума мы наблюдаем в одинаковой мере и у великих практиков, и у великих теоретиков. Кроме того, если устанавливать градации деятельности по трудности и сложности требований, предъявляемых уму, то придется признать, что с точки зрения многообразия, а иногда и внутренней противоречивости интеллектуальных задач, а также жесткости условий, в которых протекает умственная работа, первые места должны занять высшие формы практической деятельности. Умственная работа ученого, строго говоря, проще, яснее, спокойнее (это не значит обязательно легче), чем умственная работа политического деятеля или полководца. Но, конечно, установление такого рода градаций — дело в значительной мере искусственное. Главное не в них, а в том, чтобы полностью осознать психологическое своеобразие и огромную сложность и важность проблемы практического мышления [127].

Более глубокая постановка вопроса о практическом уме, преодолевающая ограниченность традиционной трактовки, наметилась в советской психологии. Это показано, например, в книге С.Л.Рубинштейна «Основы общей психологии». Касаясь вопроса о «мыслительных операциях, непосредственно включенных в ход практического действенного разрешения задачи», автор указывает, что эти операции выдвигают некоторые «специфические требования, отличные от требований, предъявляемых задачей при обобщенном теоретическом мышлении». Например, они требуют «более изощренной наблюдательности и внимания к отдельным, частным деталям, предполагая умение использовать для разрешения задачи в частном случае то особенное и единичное в данной проблемной ситуации, что не входит полностью и без остатка в теоретическое обобщение; они требуют также умения быстро переходить от размышления к действию и обратно» [110, с. 308]; Подчеркнуто Б.Тепловым). Здесь отмечен ряд особенностей, действительно характерных для практического ума, но этот перечень далеко не полный. Вопрос о практическом уме только еще ставится в психологии, и путь к его разрешению лежит через детальное изучение особенностей умственной работы человека в различных конкретных областях практической деятельности [127].

Моделирование процесса развития конструктивно-логического мышления у студента при обучении программированию

Моделирование исследуемой разновидности мышления целесообразно начать с обсуждения механизмов детально исследованного другого вида практического мышления - оперативного мышления, под которым в [29, 118] понимают такой процесс решения практических задач, который приводит к созданию у человека модели предполагаемой совокупности действий (плана операции) с реальными объектами и процессами.

Приведенное определение оперативного мышления ограничивает указанную психическую деятельность от других проявлений интеллекта: а) от других видов операциональных умственных действий эту деятельность отличает то, что ее содержанием является процесс решения задач; б) от проявлений творческого мышления оперативное мышление отличается тем, что оно непосредственно вплетено в практическую деятельность; в) от решения задач (например, математических), требующих создания плана операций, оперативную интеллектуальную деятельность отличает то, что в ней план предстоящей операции относится не к символам и знакам, а к реальным объектам или процессам.

Экспериментальное исследование оперативного мышления в [27-29, 105] было направлено на анализ конкретных видов оперативного труда, и позволило установить, что мышление в этих случаях также порождается проблемной ситуацией, то есть конфликтом между условиями и требованиями задачи. В этом сходство оперативного мышления с творческим мышлением. Однако в ходе изучения мышления, вплетенного в трудовую деятельность, были выявлены специфические особенности проблемной ситуации, характерные именно для оперативного мышления. Эти особенности определили требования к экспериментальным задачам, с помощью которых исследовалось оперативное мышление. Требования состояли в следующем: а) экспериментальная задача должна была решаться за совокупностью шагов; б) действия по решению задачи (последовательные шаги) должны быть простыми и не требовать от испытуемого специальных знаний; в) задача должна была допускать возможность варьирования проблемной ситуации, каждый вариант оценивался по степени сложности.

Исследователями был выявлен ряд процессов оперативного мышления, таких как, структурирование (образование более крупных единиц действия на основе связывания элементов ситуации между собой), динамическое узнавание (узнавание в исходной проблемной ситуации составных частей конечной ситуации), обобщение и формирование алгоритмов решения задач. Уже при решении первой задачи человек стремится выработать такие принципы и правила, которые облегчили бы ему решение задач в последующем. Сложность состоит в том, что разные задачи имеют разную сложность, различную организацию элементов, различную степень нарушения их порядка по сравнению с требуемой ситуацией – все это и определяет характер совершаемых обобщений. По мере возрастания сложности задачи (как было в указанных опытах) мыслительная деятельность человека направляется на все новые свойства элементов, а это ведет к пересмотру уже выработанных принципов, к замене их новыми, более адекватных выявленным свойствам [28, С.98].

Эти компоненты процесса оперативного мышления, по мнению авторов [29], лежат в основе деятельности, приводящей к сокращению перебора попыток решения, которое «отличает человеческое мышление от работы существующих счетно-решающих машин».

Следуя за авторами [151], вернемся к понятию алгоритма и рассмотрим его в методологическом аспекте.

«Алгоритм может приобретать две формы – идеальную и знаковую. Идеальная форма является отображением ментального образа алгоритма в ментальном пространстве человека, носителем семантического значения алгоритма. Знаковая форма есть своеобразной промежуточной формой и служит для передачи алгоритма от конструктора алгоритма к его исполнителю, а также для сохранения алгоритма для последующего использования. Следовательно, знаковая форма необходима, во-первых, для устранения семантической разницы в интерпретации алгоритма конструктором и исполнителем а, во-вторых, для сохранения или передачи алгоритма для последующего применения» [151].

И далее: «Понятно, что прежде чем научиться конструировать, составлять алгоритмы, и через них порождать алгоритмические процессы и руководить ими, нужно понять закономерности, которые имеют место в самих этих процессах. Но в большинстве случаев алгоритмический процесс наблюдать невозможно в результате его непосредственной недоступности для человека (например, за счет

скорости протекания этого процесса в компьютере и его реализации на аппаратном уровне, внешнем по отношению к человеку). О таком процессе можно делать выводы лишь по результатам (промежуточными и конечными). Такие результаты не могут дать адекватной информации о характере протекания и структуре собственно процесса. Гомоморфным алгоритмическому процессу есть алгоритм в знаковой форме. Гомоморфность здесь означает, что он не вполне однозначно соответствует процессу, но воспроизводит его существенные шаги, то есть является свернутой формой процесса. Таким образом, знаковая форма является самостоятельным объектом, который реально существует и замещает другой реальный объект – алгоритмический процесс. При этом выполняется условие: алгоритм не совпадает полностью с соответствующим ему алгоритмическим процессом, но исследование алгоритма дает полную информацию о протекании процесса. Поэтому знаковую форму алгоритма можно назвать моделью алгоритмического процесса. Заметим, что если принять такую точку зрения, то, по отношению к описанию информационной модели, алгоритм выступает в качестве метамодели (как знаковая модель, используемая для описания другой модели – информационной)).».

Признание знаковой формы алгоритма как определенной модели позволяет распространить на нее общие методы работы с моделями, т.е. алгоритмизация является моделированием алгоритмических процессов. Под моделированием будем понимать «метод опосредствованного практического и теоретического оперирования объектом, при котором исследуется непосредственно не сам объект, а используется вспомогательная искусственная или природная система, что находится в определенном объективном соответствии с познаваемым объектом и способна замещать его на определенных этапах познания, что дает при его исследовании в конечном счете информацию о самом моделируемом объекте» [94, с.42].

«Алгоритм» - продолжает А.В.Копаев – «рассматривается в качестве определенной сущности, которая может приобретать идеальную и материальную форму. Мы считаем, что идеальная форма является носителем семантического значения алгоритма (то есть, отображением его ментального образа в структуре ментального опыта человека или структуре памяти исполнителя). Поэтому такого рода идеальное, семантическое представление алгоритма единственное (ведь его изменение приводит к созданию уже другого алгоритма).

Знаковая же форма опосредствуется определенными знаковыми представлениями, которые определяются особенностями синтаксических правил в соответствующих им знаковых системах. И это объясняет наличие многих видов знакового представления алгоритма. Каждый такой вид имеет определенное своеобразие и освещает алгоритм с той или иной стороны. Следовательно, материальная форма определенными средствами (знаковыми или символическими системами) отображает идеальную форму. Но и идеальная форма тоже должна определенным образом отображать форму знаковую. Знаковая форма после восприятия человеком трансформируется в соответствующий ментальный образ алгоритма, то есть во время исследования или разработки алгоритма ученик всегда строит определенный ментальный образ алгоритма как объекта, на который направлена его деятельность. При этом процесс восприятия сопровождается пониманием (или непониманием) и может зависеть от понятности или непонятности самого объекта, что воспринимается.».

Во внешнем плане алгоритм, представленный посредством определенной знаковой системы, воспринимается и фиксируется как системно организованное изображение (схема, план, граф). А.В.Копаев считает [151], что, чем большее это изображение приближается по своей структуре к ментальному образу алгоритма, который мысленно создает ученик, тем более оно будет способствовать понятности собственно алгоритма. В большинстве случаев в практике обучения алгоритмике используются текстовые представления алгоритмов. Это объясняется с психологической точки зрения тем, что знаковые модели являются моделями вербальными, теоретическими и уже это само по себе будет способствовать формированию теоретического мышления. Но сами по себе ментальные модели, которые формируются в ученика, даже если они и отображают теоретические понятия, не могут быть оторваны от чувственного. Современная педагогическая психология указывает, что успешность обучения в значительной степени зависит не только от использования самих моделей, но еще и от характера их использования при решении учебных задач. Поэтому мы видим одно из основных значений знаковой формы алгоритма с познавательной точки зрения именно в том, что она является своеобразной наглядностью. Она связывает теоретические абстракции, воплощением которых есть алгоритм в идеальной форме, и их

наглядные эквиваленты. Такого рода связь всегда существует. Это связано, прежде всего, с тем, что понятийное мышление невозможно без наглядного. И эта связь так или иначе давно используется в дидактике, где она нашла воплощение в форме принципа наглядности обучения. В соответствии с им обучение должно опираться на применение конкретных образов, используемых учениками. Конечно, под конкретностью понимается не единичное, а конкретная форма представления алгоритма как всеобщего. При таком подходе назначения знаковой формы алгоритма видится нами не в обогащении чувственного опыта ребенка, а как чувственная опора для формирования теоретических понятий в связи со специальной дидактической задачей – формированием алгоритмического стиля мышления. Знаковая форма алгоритма в качестве модели выступает и как продукт, и как средство осуществления теоретической деятельности через наглядно-образные формы. Это позволяет утверждать, что алгоритмы-модели являются своеобразным сочетанием чувственного и рационального в познании. Этот вывод важен, потому что позволяет по-новому взглянуть на дидактическое значение и применение представлений алгоритмов. При правильном выборе алгоритмического представления можно развивать не только теоретическое мышление (свойственное алгоритмическому стилю), но и наглядно-образное. Например, можно с достаточной достоверностью прогнозировать, что применение графическо-символьных представлений алгоритмов будет способствовать обучению алгоритмике детей, в которых преобладает наглядно-образный компонент над аналитическим. Также А.В. Копяев отмечает в [151], что алгоритмы, как своеобразная форма наглядного представления процессов, используются не просто как иллюстрация некоторого положения, но и как отображение активных моделей. Алгоритмы являются не просто иллюстративными моделями, которые односторонне воспроизводят алгоритмические процессы. Они являются также моделями проектирующими, то есть такими, что порождают эти процессы, позволяют их организовывать и реорганизовывать, полностью предугадывая их поведение. Таким образом, алгоритмизация рассматривается нами как специфическая познавательно-проектирующая деятельность. Алгоритмизация – это не просто знание алгоритмов и их воссоздание. Это, прежде всего, овладение общими способами действий, приемами, средствами создания и применения алгоритмов.

Авторы проекта «Алгоритмика» считают, что с точки зрения обучения алгоритмике в школе наиболее пригодно понятие алгоритма, используемое в прикладной теории алгоритмов. Эмпирическое понятие алгоритма, с одной стороны, является формальным, что отличает его от «размытого», а, с другой стороны, является понятным, простым для применения в сравнении с его математическим толкованием. Поэтому дальше в работе, используя термин «алгоритм», мы будем понимать его с точки зрения эмпирического понятия.

Развивающее обучение алгоритмике. В данный момент времени в педагогической практике наблюдается повышенный интерес к так называемому развивающему обучению, что, как известно, предусматривает целенаправленное, организованное развитие психических качеств ученика [17]. Анализ и проектирование любой методической системы обучения с точки зрения принципов развивающего обучения возможны лишь при условии глубокого психолого-педагогического осмысления проблем, которые встают на таком пути.

Система обучения любого предмета, в частности и информатики, сложна. Сложность свойственная как информатике в целом, так и каждому ее отдельному разделу. Поэтому при создании методической системы обучения алгоритмике можно применять различные подходы, которые основываются на разных точках зрения, как на саму алгоритмику, так и на ее предмет. Придерживаясь мысли о том, что процесс обучения — это, прежде всего, деятельность ученика, который является главной фигурой этого процесса, — мы считаем, что психолого-педагогический анализ методической системы обучения алгоритмике и проектирования в дальнейшем на его основе методической системы, должны проводиться с учетом всех видов деятельности, в которые вовлечен ученик. Концепция развивающего обучения основывается на социокультурной (Выготский Л.С.) и деятельностной психологических теориях (Леонтьев А.Н., Рубинштейн С.Л.). С точки зрения этих теорий обучение является процессом организованного специальным образом освоения учениками опыта общественно-исторической практики, культурных ценностей, созданных предшествующими поколениями. И такое освоение невозможно никаким иным образом, кроме как через осуществление самой *предметно-специфической деятельности*.

Таким образом, основной целью учебной деятельности и ее продуктом является система умений, навыков и знаний, которая позволяет уверенно (компетентно) осуществлять определенную предметно-специфическую деятельность. Причем, потенциальная и актуальная возможность осуществления деятельности личностью является основным критерием освоения ею этой деятельности. Поэтому подход к обучению с развивающей, деятельностной позиции с необходимостью ставит вопрос о психологическом анализе структуры деятельности и ее компонентов, в частности, предмета, средств и способов применения этих средств к преобразованию предмета.

При обучении алгоритмике предметно-специфической является алгоритмическая деятельность. Под алгоритмической деятельностью мы понимаем деятельность, целью которой является создание, понимание и преобразование алгоритма, который является и предметом, и непосредственным продуктом этой деятельности. Таким образом, встает вопрос о рассмотрении понятия алгоритма именно как предмета алгоритмической деятельности. Позиция авторов проекта «Алгоритмика» в том, что при обучении алгоритмике целесообразно рассматривать алгоритм как модель алгоритмического процесса [6]. Алгоритмизация при таком подходе рассматривается как моделирование алгоритмических процессов.

Когда речь заходит о модели, то имеют в виду, как правило, познавательную функцию моделей [151]. Между тем, можно выделить модели, которые не только позволяют познать другой предмет, но и проектируют, задают его. Таким образом, «модель может служить для достижения одной из двух основных целей: или описательной, если модель служит для объяснения и (или) лучшего понимания объекта, или предписательной, когда модель позволяет предусмотреть и (или) воссоздать характеристики объекта, которые определяют его поведение» [151]. Примерами таких моделей является план дома, схема застройки нового микрорайона, чертеж изделия, которого еще не существует в материальной форме, и тому подобное. Отметим, что такие модели в большинстве случаев являются знаково-символьными. Они проектируют еще не существующую реальность и проектируют ее с вполне заданными характеристиками. Специфика алгоритмов как моделей в том, что они есть одновременно и описательными, и предписательными (проектировочными). Р.Шеннон отмечает: «модель, пригодная для цели разработки системы, также должна и объяснять ее, но очевидно, что модели, которые создаются

исключительно для объяснения, часто не отвечают даже своему непосредственному назначению» [135, с.19]. А.В.Копаев считает, что высказанное Р.Шенноном мнение, позволяет объяснить ситуацию, которая сложилась в современной практике преподавания алгоритмики в школе. Ведь понятие алгоритма, в том виде в котором оно применяется, является описательным в большинстве случаев. Понятие «системы предписаний», «сообщения о том, как необходимо решать задачу» или «конечной последовательности действий», по нашему мнению, никаким образом не способствуют пониманию сущности алгоритмов и алгоритмических процессов, как искусственной реальности, что проектируется, исследуется и модифицируется самим учеником.

Таким образом, если истолковывать алгоритм как описательно-проектировочную знаково-символьную модель, то средствами алгоритмической деятельности являются алгоритмические представления (языки), а способы их применения для построения и модификации алгоритмов являются способами осуществления самой этой деятельности. По мнению А.В.Копаева такой подход дает возможность сравнивать эффективность алгоритмических нотаций с точки зрения эффективности их применения для проектирования или понимания алгоритмов. Также мы согласны с Копаевым А.В. в том, что интерпретация алгоритмов как моделей алгоритмических процессов, а алгоритмизации как моделирования этих процессов, позволит применить в системе обучения алгоритмики понятия и методы, которые приняты в общей теории моделей и систем, что внесет определенную системность и большую научную строгость в саму методику обучения, а термины «системное» и «операционное» мышление наполнит конкретным содержанием.

«Под *оперативным мышлением* понимают процесс решения практических задач, который приводит к созданию у человека мыслительной модели предполагаемой совокупности действий – плана операций – с реальными объектами и процессами» [51]. Характерными признаками оперативного мышления есть структурирование информации, динамическое узнавание ситуации, формирование алгоритмов принятия решения и собственно выполнение этого решения. Еще на раннем этапе развития ЭВМ и систем управления на их основе авторы [52] писали: «Мы считаем, что операторская деятельность имеет алгоритмический характер, дополненный элементами оптимизации и эвристики при принятии

решений. Поэтому формирование и развитие алгоритмического стиля мышления служит основой для формирования мышления оперативного.».

В свою очередь, на понятие алгоритмического стиля мышления влияют особенности оперативного мышления. Среди таких особенностей выделяется, прежде всего, его практическая направленность. Применяя оперативное мышление при решении разнообразных задач реальной жизни, человек должен спланировать не только самые действия, но и очертить ресурсы (информационные, технические), что при этом используются. В каждом таком случае нужно решать диалектическую альтернативу определенных факторов или факторов эффективности – временных, пространственных, материальных и т.п. При этом возникает проблема выбора наиболее целесообразного, оптимального плана действий в зависимости от имеющейся ситуации (то есть выбора по определенным критериям, которые диктуются конкретной динамической ситуацией). Расширение понятия алгоритмического стиля мышления с учетом ориентации не просто на решение задач, а нахождение наиболее оптимального решения при заданных условиях, применение наиболее оптимального алгоритма из доступных, позволяет наполнить задачи, которые рассматриваются в курсе школьной информатики, практическим содержанием. Стиль мышления, который формируется под влиянием решения такого рода задач, может быть, сознательно или даже подсознательно, использован человеком во время решения не только теоретических, а и практических задач, в основе которых лежит создание эффективного плана действий. Поэтому, мы считаем, что алгоритмический подход важен не только при обучении алгоритмике или программированию, а является одним из главных в обучении информатики вообще. Ведь еще А.П.Ершов считал, что "информатика – это наука о правилах целенаправленной деятельности". И относительно алгоритмики это высказывание является наиболее справедливым. Именно алгоритмика и алгоритмический стиль мышления, которое формируется и развивается при обучении алгоритмики, есть необходимой базой для усвоения учениками в будущем, как элементов программирования, так и технологических компонентов информатики.

1.1. Подход к моделированию процесса развития конструктивно-логического мышления у студентов

Основные положения предлагаемого подхода к развивающему обучению программированию состоят в следующем.

1. Профессиональная деятельность работников, как в большинстве отраслей современной промышленности, так и во всех сферах общественного производства и общественной жизни, требует поиска или конструирования решения возникающих задач за ограниченное время и с ограниченными ресурсами. Это требует от исполнителя определенного склада мышления, который можно охарактеризовать как конструктивный и, одновременно, подчиняющийся строгим формальным, логическим законам и правилам, инструкциям. Будем называть этот склад ума, вид мыслительной деятельности «конструктивно-логическим мышлением», сокращенно КЛМ.

Все более усложняется и становится все более строгой регламентация всех сторон жизни человека и деятельности работника, протекающих по законам, правилам и инструкциям, даже в быту, например, человек всюду сталкивается с различными приборами, управление которыми организовано и визуализировано в форме разветвленных, динамических, многоуровневых, древовидных меню, которые приближаются к так называемым предписаниям алгоритмического типа, близким к математическим объектам – алгоритмам. Отсюда возникает насущная потребность формировать у будущих работников стиль мышления, который можно назвать «алгоритмическим», сокращенно АСМ.

На наш взгляд, конструктивно-логическое мышление активно используется всеми специалистами, управляющими работой организационно-экономических и социальных систем. Например, преподаватель, управляющий учебной деятельностью учеников или студентов, во многом опирается на конструктивно-логическое мышление, как собственное, так и обучающихся.

Алгоритмы решения задач и программы для ЭВМ, составленные на их основе, по определению являются конструктивными объектами, так что освоение основ алгоритмизации, владение хотя бы одним языком программирования и умение записать с помощью этого языка программу, решающую поставленную задачу, все это, на наш взгляд, наиболее эффективный путь к развитию комплекса КЛМ-АСМ у

обучаемого, даже в случае, когда речь не идет о подготовке программистов-профессионалов.

Применительно к программированию конструктивно-логическое мышление (КЛМ) – это сложный динамический комплекс основных видов мышления, компоненты которого активизируются в различных сочетаниях на разных этапах программирования для решения на ЭВМ.

2. Характеристика различных возрастных периодов в развитии конструктивно-логического мышления.

На наш взгляд, вся сознательная жизнь человека в современном обществе через обучение и воспитание формирует конструктивность и логичность мышления, но в разные возрастные периоды имеет место определенная чувствительность к различным методам и средствам формирования и развития этих качеств.

Правильно воздействуя на мышление обучаемого в указанные периоды развития можно достичь существенных результатов в развитии КЛМ. Понимая под КЛМ – мыслительные действия, направленные на анализ и конструирование (планирование) целенаправленной деятельности и организация использования ограниченных ресурсов (материальных, пространственно-временных) в процессе решения задачи, можно выделить возрастные периоды и виды деятельности, формирующие это мышление:

1) ребенок – в игре, в конструктивной деятельности (здесь анкетуем и узнаем, чем занимался в раннем детстве, т.е. в первый сенситивный период);

2) ученик – в учебной деятельности, на уроках физики, математики, информатики (здесь анкетуем и узнаем, когда приобщился к ЭВМ и программированию, компьютерный и программистский стаж – использован ли второй сенситивный период);

3) в дальнейшем, студент – на занятиях, например, по информатике и программированию (тестируем и оцениваем уровень развития КЛМ после 1-го полугодия, после второго полугодия – смотрим динамику КЛМ-АСМ);

4) специалист – при решении задач предметной области (Про) (уровень развития КЛМ-АСМ существенно влияет на успешность решения профессиональных задач).

Например, начало развития наглядного-действенного мышления и освоения игровой деятельности младшим дошкольником является и началом формирования КЛМ. У ребенка КЛМ как вид мышления формируется, начиная с раннего детства (2-3 года), когда начинается

освоение предметной деятельности, и освоении игровой деятельности в дошкольном возрасте, как и учебно-познавательной – в младшем школьном возрасте.

Наиболее выраженный сенситивный период это 11-12 лет, т.е. переход в 5-6 классы средней школы - именно в это время начинается системное изучение естественных наук.

Существенный скачок в развитии исследуемого качества связан с началом изучения основ наук в средних классах школы. В этот период многие школьники, особенно мальчики, впервые приобщаются к информатике.

В период окончания средней школы, в старших классах, как и в системе среднего профессионального образования при подготовке к трудовой деятельности в избранной сфере, уже два десятилетия изучается школьный курс «Основы информатики и вычислительной техники» - здесь комплекс КЛМ-АСМ формируется через обучение основам информатики и программирования задач для ЭВМ.

В период завершения обучения в средней школе и начала профессиональной подготовки, когда освоены начала наук и начата подготовка к трудовой деятельности в избранной сфере, появляется возможность формировать комплекс КЛМ-АСМ через обучение программированию задач для ЭВМ.

3. На наш взгляд, в конструктивно-логической деятельности участвуют все виды мышления, но степень участия каждого вида мышления на различных этапах программирования задачи является темой отдельного исследования, поэтому в данной работе предлагается оценивать относительные изменения уровня развития конструктивно-логического мышления у студента через его успешность в конкретно-специфической деятельности, то есть в изучении программирования.

4. С одной стороны, современная математика (особенно ее прикладные разделы), и современное программирование, являются весьма яркими и значимыми областями приложения КЛМ-АСМ, а, с другой стороны, именно обучение математике и программированию, на наш взгляд, является одним из наиболее продуктивных способов развития КЛМ и АСМ, или в целом комплекса КЛМ-АСМ. Составление алгоритмов на аудиторных занятиях, разработка программ на практических занятиях в компьютерном классе, с доведением до законченного рабочего продукта, в отличие, например, от обучения строительному делу или другим инженерным

специальностям, сравнительно проще организуется и требует относительно небольших затрат, т.е. являются доступными для студентов. В то же время, обучение основам алгоритмизации и программирования студентов (а через них КЛМ-АСМ) любых естественнонаучных, экономических и инженерных специальностей позволяет сформировать дополнительный базис для обучения любой специальности, в том числе специальностей, предъявляющих особые требования к будущему работнику.

Особое значение развитие КЛМ-АСМ, как и знание закономерностей и способов развития этого комплекса, имеет для студентов - будущих педагогов математиков и информатиков, ведь именно они будут призваны в школу, чтобы развивать у учащихся этот психоконструкт, который необходим любому выпускнику школы, независимо от дальнейшего выбора профессии или специальности обучения в вузе.

Алгоритмы решения задач и программы для ЭВМ, составленные на их основе, являются конструктивными объектами, так что освоение основ алгоритмизации, знание хотя бы одного языка программирования и умение записать с помощью этого языка программу, решающую поставленную задачу, все это, на наш взгляд, весьма эффективный путь развития комплекса КЛМ-АСМ у обучаемого, даже в случае, когда речь не идет о подготовке программистов-профессионалов.

5. На сегодняшний день в разделах информатики, посвященных алгоритмам и алгоритмизации, имеется достаточное количество учебной и методической литературы для того, чтобы учить основам алгоритмизации студентов любых специальностей любого уровня образования.

В области языков программирования достаточно привести пример языка программирования (ЯП) Паскаль (Pascal), созданного швейцарским программистом и педагогом Никлаусом Виртом еще в 1970 году, и с тех пор являющегося основным инструментом обучения началам программирования, в том числе и будущих программистов. Для этого языка созданы доступные и удобные турбо-среды, не требующие особых вычислительных ресурсов и ресурсов памяти, а это значит, что практиковаться в Паскале может любой учащийся, имеющий доступ даже к весьма простым персональным компьютерам ранних поколений

Особыми качествами ЯП Паскаль являются:

а) развитая и гибкая система типов данных, дающая возможность программисту максимально точно отобразить структуру исходных данных задачи на структуры в памяти ЭВМ;

б) возможность реализовать задачу, предварительно разбитую на подзадачи, в виде совокупности независимых модулей, допускающих раздельную разработку.

6. Модель программирования и обучения программированию. Представленная в этом разделе модель поведения программиста, описанная в [167], основана на экспериментальных результатах и может быть полезна для формулирования гипотез.

Всякая модель поведения программиста должна учитывать, по крайней мере, следующие описанные задачи программирования:

- 1) составление - написание программы;
- 2) понимание - понимание заданной проблемы;
- 3) отладка - нахождение ошибок в заданной программе;
- 4) тестирование - проверка того, что программа удовлетворяет своему назначению;
- 5) модификация - изменения заданной программы с тем, чтобы она была предназначена для выполнения другого задания;
- 6) обучение - овладение новыми навыками и знаниями программирования.

Кроме того, умозрительная модель должна быть пригодна для описания этих задач в терминах *мысленных структур*, которые программист хранит или должен хранить в своей памяти, и *мыслительных процессов*, применяемых при использовании этих знаний или при их расширении.

В исследовании [137], основанном на *информационном подходе к психологии* обучения, запоминания и решения задач, была предложена схема на основе компонентов памяти, используемых в работе программиста.

7. В отличие от технократического подхода Б.Шнейдермана [137] к описанию и моделированию деятельности программиста и деятельности по обучению программированию, мы считаем, что при обучении алгоритмизации и программированию с опорой на общие дидактические принципы у обучаемых интенсивно развивается когнитивный комплекс КЛМ-АСМ, а уровень развития этого комплекса можно достоверно оценить через оценку уровня

компетенции обучаемых в области алгоритмов и программирования.

Обучать алгоритмизации и программированию, основываясь на общедидактических принципах системности, научности, наглядности, активности и других – это структурирование данных и действий, придание этим процедурам наглядного графического характера; активность как работа обучаемого в роли Исполнителя программы, например, в режиме пошаговой отладки программы.

Мы (как и специалисты «Алгоритмики») предлагаем использовать традиционный для отечественной дидактики и педагогической психологии деятельностный подход, проблемное обучение с опорой на систему дидактических принципов, адаптированных к обучению студентов программированию.

Ниже приводится характеристика общедидактических принципов применительно к обучению программированию.

Дидактические принципы, или принципы обучения,— исходные положения теории обучения. Они определяются целями обучения и воспитания, потребностями общественного развития, особенностями учебной деятельности обучаемых, выявляемыми на основе данных психологии и социологии. В свою очередь принципы дидактики определяют методы обучения. Последние строятся так, чтобы обеспечить наилучшим образом реализацию принципов дидактики.

Дидактические принципы взаимно связаны и образуют систему принципов (разумеется, в различных педагогических изданиях встречаются различные варианты эквивалентных систем).

Примем за основу систему из шести принципов, использованную, например, А.А.Столяром при разработке теории обучения математике в средней школе [114]: научность в обучении; сознательность усвоения; активность обучаемых; наглядность обучения; прочность знаний; индивидуальный подход.

Рассмотрим вопрос о реализации каждого из этих принципов, а также связь между ними в обучении программированию. Цель – выявить недостатки традиционной методики в реализации дидактических принципов и определить направление усовершенствования методов обучения с целью достижения лучшего их применения в учебном процессе.

Научность в обучении. Принцип научности состоит в том, что образовательный материал, составляющий содержание обучения в вузе, должен (в возможной мере) соответствовать уровню

современной науки, преподноситься студентам в определенной (дидактической) *системе*, отражающей научную систему, в определенной *последовательности*, сохраняющей связи понятий, тем, разделов внутри каждого предмета, а также межпредметные связи.

Иногда в педагогической литературе рассматриваемый принцип именуется принципом «научности, систематичности и последовательности» в обучении, но можно называть кратко принципом научности в обучении, так как научность включает и систематичность, и последовательность.

Разумеется, механический перенос системы науки даже в вузовское обучение сложен. Как отмечал А.А.Столяр, научная система подвергается специальной (дидактической) обработке, в результате которой получается дидактическая система — учебный предмет.

Дидактическая система характеризуется следующими особенностями:

а) отражает (но не воспроизводит в точности) систему науки, сохраняя по мере возможности в общих чертах присущую ей логику и систему знаний;

б) в ней последующее опирается на предыдущее;

в) вся система знаний по предмету, порядок расположения материала по семестрам и годам обучения соответствует психологическим особенностям и возможностям обучаемых и способствует их дальнейшему более быстрому развитию;

г) раскрывает внутреннюю связь между научными понятиями, закономерностями, а также связь данного предмета с другими предметами.

Последовательное расположение материала является необходимым условием систематичности. Каждая система характеризуется определенной последовательностью расположения материала, и сравнительный анализ различных систем изложения одного и того же материала сводится к сравнению соответствующих последовательностей расположения материала и идей, лежащих в их основе.

На наш взгляд, принцип научности при обучении программированию реализуется при условиях:

а) строгого соблюдения технологии решения задачи на ЭВМ, последовательности выполнения этапов;

б) главный акцент в обучении, как и в деятельности по программированию, должен делаться на этапах анализа и структурирования задачи и последующей алгоритмизации, на которых структурируются задача (ситуация), ресурсы (ЭВМ) и действия (алгоритм);

в) использования в решении задачи на ЭВМ принципа структурирования задачи, алгоритма и структур данных, что является конкретным воплощением общего принципа структурирования проблемы (ситуации управления) и ресурсов для ее решения; этап анализа структуры задачи и ее решения должен быть отрефлексирован и реализован обучаемым явно, наглядно, т.е. с визуализацией и словесным комментарием - при наглядной пошаговой отладке программ в среде программирования с использованием тестовых данных решение задачи на ЭВМ становится предельно наглядным, обеспечивая реализацию данного принципа.

Наглядность обучения. Наглядное обучение, по словам К.Д.Ушинского, такое обучение, которое строится не на отвлеченных представлениях и словах, а на конкретных образах, непосредственно воспринятых ребенком.

Наглядность особенно важна в обучении программированию ввиду того, что здесь требуется достижение более высокой ступени абстракции, чем в обучении другим предметам, а она содействует развитию абстрактного мышления.

Из различных видов наглядности — натуральной, изобразительной, символической — широкое применение в обучении программированию находит символическая наглядность (чертежи, графики, схемы, таблицы).

Средство символической наглядности представляет собой условную знаковую систему, с помощью которой изучаемая сторона предметов, явлений, процессов отделяется от прочих свойств и представляется в чистом виде. Например, обычно используемая ГСА является средством наглядности, так как отделяет изучаемую структуру действий по решению задачи от прочих ее свойств и представляет ее в чистом виде.

Однако не всем сразу ясна символическая наглядность. Представляя собой условную знаковую систему, символическая наглядность по существу является своеобразным языком и, как всякий язык, должна специально изучаться, чтобы стать понятной. Только в

таком случае символическая наглядность будет эффективным средством обучения.

Выясним роль наглядности в процессе формирования понятий алгоритмизации и программирования. Формирование понятий — сложный психологический процесс, начинающийся с образования простейших форм познания - ощущений - и протекающий по следующей схеме: ощущения - восприятие - представление - понятие.

Обычно различают две ступени этого процесса: *чувственную*, состоящую в образовании ощущений, восприятия и представления, и *логическую*, заключающуюся в переходе от представления к понятию с помощью обобщения и абстрагирования.

Чувственная ступень в процессе формирования понятий соответствует первому этапу пути познания, т.е. «живому созерцанию», и поэтому осуществление этой ступени требует применения средств наглядности.

Использование наглядности в процессе формирования понятий будет эффективным, если оно ориентирует обучаемых на обобщение и абстрагирование существенных признаков формируемого понятия.

В этом единстве противоположностей — диалектика развития понятий. Приведенный пример показывает, как применение наглядного материала в процессе формирования понятий должно быть направлено на развитие обобщающей и абстрагирующей деятельности обучаемых.

Процесс формирования понятий программирования не всегда протекает по приведенной схеме. В частности, когда формируемое понятие связано в той или иной форме с категорией, то чувственная ступень отсутствует, так как мы не в состоянии воспринимать бесконечное (ни в какой форме), и наглядность из средства, способствующего формированию понятия, становится тормозящим фактором.

Задача, решенная на уровне блок-схемы, иллюстрирует идею и алгоритм решения, но требует перевода на язык, понятный машине, т.е. алгоритм должен быть переведен с языка плоского двумерного рисунка-графа на одномерный язык последовательности символов и конструкций алгоритмического языка. Для обучаемых с преимущественным развитием геометрического или алгебраического типов математического мышления такой перевод оказывается нетривиальной задачей, а наглядность, на первых порах, будет тормозящим фактором.

Этот пример подтверждает выводы наших психологов о том, что восприятие наглядного материала в силу объективных особенностей этого материала может играть не только положительную, но и отрицательную роль. Наглядность может также играть отрицательную роль в силу неправильного применения в процессе обучения [143].

Эффективность наглядного обучения зависит от правильного выбора средств наглядности и от их правильного применения в процессе обучения.

Логико-математический язык наряду с другими «педагогическими» качествами, о которых мы будем говорить дальше, обладает и «наглядностью». Речь идет о новом виде наглядности, который мы разьясим здесь лишь на нескольких примерах. Однако само использование логико-математического языка в нашем дальнейшем изложении неоднократно покажет эту наглядность «в работе».

Так же как чертеж обнажает геометрическую форму (структуру) предмета и представляет ее в чистом виде, символическая запись предложений и рассуждений на логико-математическом языке, граф-схемы структуры задачи и ее подзадач, ГСА (граф-схем алгоритмов) обнажает их логическую форму (структуру) и представляет ее в чистом виде. Поэтому она также служит средством наглядности при решении соответствующих педагогических задач (разьяснение точного смысла высказываний сложной структуры, выявление различия в понятиях, отыскание пути доказательства и т. д.).

При наглядной пошаговой отладке программ в среде программирования с использованием тестовых данных решение задачи на ЭВМ становится предельно наглядным, обеспечивая реализацию данного принципа.

Сознательность усвоения. Сознательность усвоения понимается как такое овладение обучающимися знаниями, которое включает глубокое понимание усвоенного и умение применять его в новых конкретных ситуациях.

Сознательное усвоение знаний принципиально исключает догматическое преподавание, результатом которого являются «формальные знания». Формализм в знаниях является противоположностью сознательности усвоения и нередко встречается в обучении программированию. Это объясняется, в частности, широким использованием формализованного языка, что создает возможность отрывать в обучении форму, язык от выраженного в нем

содержания вследствие неправильного, одностороннего изучения этого языка. Такой отрыв и является источником формализма в знаниях. Формальные знания характеризуются тем, что заучивается и запоминается внешнее, формальное, символическое выражение содержательного математического факта, сам же этот факт либо вовсе отсутствует в сознании, либо присутствует вне всякой связи со своим формальным выражением, никак не ассоциируется с ним в представлении учащегося.

Можно утверждать, что информатика, в целом, и программирование, в частности, находятся в приближенной и выигрышной позиции по отношению к практике. Тем не менее, можно привести несколько примеров формальных знаний:

1. Опыт показывает, что школьники, да и студенты, изучающие программирование, систематически пренебрегают первыми этапами решения задачи – анализом структуры задачи, формализацией (построение математической модели и выбор метода решения) и алгоритмизацией (построение алгоритма решения на основе выбранного метода). Особенно вызывающе выглядит отторжение второго – нередко студенты рисуют алгоритм по уже готовой программе, то есть, образно говоря, рисуют чертеж уже построенного дома.

2. Знание циклической действий и отдельных типов циклов остается формальным, если студент не умеет преобразовать один тип цикла в другой при сохранении результата – этот тип заданий является своеобразной «лакмусовой бумажкой», позволяющей, в случае успешного выполнения, присваивать студенту неформальный поощрительный статус «программиста».

В каждом из приведенных примеров усматривается признак формальных знаний, определенный отрыв внешней формы (языка; чертежа) от выраженного в ней содержания, запоминание формы и непонимание содержания. Такие «знания» учащиеся не в состоянии применять в новых конкретных ситуациях, когда одно и то же содержание облечено в новую форму, и часто ошибаются, когда различные содержания имеют сходную форму.

Важная педагогическая проблема состоит не только в том, чтобы устранять формализм в знаниях обучаемых (хотя и эта задача возникает и должна решаться), но и в том, чтобы строить обучение, не порождающее формализма [114].

Необходимым компонентом такого обучения является правильное изучение языка алгоритма и программ (терминологии и символики), достигающее понимания его семантики.

На наш взгляд, программирование как вид деятельности и вид познавательной учебной деятельности студента, по определению, в силу своей практической направленности предопределяет сознательное отношение обучаемого.

При наглядной пошаговой отладке программ в среде программирования с использованием тестовых данных решение задачи на ЭВМ становится предельно наглядным, не совместимым с чисто формальной синтаксической правильностью текста программы, обеспечивая реализацию данного принципа.

Активность обучаемых. Сознательность усвоения предполагает активность обучаемых в процессе обучения. Без активной мыслительной деятельности невозможно сознательное усвоение знаний.

Педагогика не может строить обучение так, чтобы у ученика осталась свобода выбора между активной мыслительной деятельностью и простым заучиванием. Она должна строить обучение программированию как *активное* обучение, основой которого служит активная мыслительная деятельность всех обучаемых [114].

Следуя А.А.Столяру, будем различать два вида активности обучаемых в процессе обучения (математике/программированию), названным им соответственно активностью в *широком и узком смысле*.

Активность в широком смысле в обучении программированию не отличается существенно от активности обучаемых в процессе обучения их другим предметам. Это вообще активная мыслительная деятельность. Активность в узком смысле — это специфическая активность, мыслительная деятельность определенной структуры, свойственная для программирования.

Если студент проявляет активность в узком смысле, то он проявляет и активность в широком смысле. Обратное же неверно. Студент может проявлять активную мыслительную деятельность вообще, но она не всегда является специфической программистской деятельностью, т.е. он может проявлять активность в широком смысле, но не проявлять при этом активности в узком смысле [143].

Проблема активного обучения программированию, хотя и не нова, остается актуальной, так как до сих пор не получила

удовлетворительного решения в широкой практике вузовского и школьного преподавания. К тому же в связи с необходимой модернизацией обучения программированию проблема активного обучения принимает и новый аспект.

В случае обучения студентов — будущих преподавателей математики информатики, нацеленных на развитие мышления обучающихся — под активностью в широком смысле мы понимаем активное и сознательное участие студента в педагогическом эксперименте, обсуждение с ним всех сторон его обучения.

На основании понимания «активного» обучения программированию как обучения деятельности, нас интересует и построение обучения, основой которого является активность в узком смысле, и возникающие при этом проблемы.

Активность обучения программированию в узкоспециальном смысле, на наш взгляд, предопределена использованием компьютеров на занятиях, тогда как в широком смысле познавательную активность студента способны вызвать: а) специально организованная самостоятельная работа; б) задачи с реальным содержанием, близким с предметной области и специальности, которую получает обучаемый. То есть будущим педагогам целесообразно давать задачи широкого плана с педагогическим контекстом, физику — задачи на моделирование природных явлений, процессов в технических системах, будущему экономисту или менеджеру — задачи с реальным экономическим содержанием, опирающиеся на их начальные профессиональные знания.

При наглядной пошаговой отладке программ в среде программирования с использованием тестовых данных из реальной практики, когда студент сам выступает соисполнителем программы, решение задачи на ЭВМ становится предельно наглядным, обеспечивая реализацию данного принципа.

Прочность знаний. Принцип прочности знаний требует, чтобы у обучаемых сохранились на длительное время систематизированные знания, умения и навыки. Это невозможно осуществить путем простого заучивания, без глубокого понимания изучаемого материала.

Таким образом, одним из необходимых условий прочности знаний является сознательность усвоения. Но сознательность усвоения обеспечивается, как мы видели выше, активной мыслительной деятельностью обучаемых. Поэтому необходимым условием

прочности знаний является приобретение их активным способом. Однако это условие еще не является достаточным.

Разрозненные, несистематизированные, не связанные общими идеями знания не могут быть прочными. Но систематичность обеспечивается научностью обучения. Таким образом, другим необходимым условием прочности знаний является научность обучения.

Научность и сознательность, вместе взятые, не являются еще достаточным условием прочности знаний. Если бы прочность была следствием научности и сознательности, то отпала бы необходимость выделения ее в качестве отдельного принципа. Для обеспечения прочности знаний необходима еще соответствующая организация обучения, учитывающая результаты исследований механизма запоминания.

В педагогике математики пользуются некоторыми общепедагогическими положениями, отражающими результаты психолого-педагогических исследований, а именно: а) запоминание находится в прямой зависимости от повторения; б) память имеет избирательный характер - запоминается преимущественно то, что для нас существенно, интересно; в) материал запоминается лучше, когда мотивом является применение его на практике; г) запоминанию способствует разбиение изучаемого материала на небольшие порции по смысловому содержанию с выделением опорных пунктов в 'форме тезисов, заголовков, вопросов; д) эмоционально окрашенный материал при прочих равных условиях запоминается лучше. К этим общепедагогическим положениям добавим лишь некоторые замечания, касающиеся обучения математике и программированию.

Прежде всего, возникает вопрос, что должен запоминать студент?

При обучении программированию важно научить студентов пользоваться справочными материалами, в том числе встроенными Help-справочниками современных турбо-систем программирования, принимая во внимание, что нередко эти фирменные программные продукты не русифицированы, а оснащены текстовой информацией на английском языке.

Повторение ранее изученного материала в связи с его применением к изучению нового материала является наиболее важным видом повторения. Оно связывает различные понятия, предложения в систему, способствующую лучшему запоминанию, как старого, так и нового материала.

Прочность знаний в алгоритмизации и программировании может быть обеспечена при многократном решении задач – составлении блок-схем алгоритмов, когда типовые алгоритмические конструкции применяются в различных сочетаниях или преобразуются друг в друга, при кодировании алгоритма на алгоритмическом языке, при наглядной пошаговой отладке программ в среде программирования.

При наглядной пошаговой отладке программ в среде программирования с использованием тестовых данных и справочных материалов различного рода решение задачи на ЭВМ становится предельно наглядным, обеспечивая реализацию данного принципа.

Индивидуальный подход. Рассматривая обучение как процесс управления, мы приходим к выводу, что для осуществления эффективного процесса обучения надо учитывать особенности обучаемого (управляемого объекта). Здесь имеются в виду особенности мышления каждого ученика, свойства его памяти, отдельных анализаторов (слух, зрение), а также его характер и воля. Без учета этих особенностей мы не можем достичь доступности обучения. Даже у обучаемых одного возраста указанные особенности весьма различны. Поэтому один и тот же материал одни учащиеся усваивают быстрее, а другие — медленнее. Это обуславливает необходимость индивидуального подхода в обучении.

Если можно было бы как-то «измерить» скорости усвоения материала по алгоритмам и программам различными учащимися, то определенно обнаружился бы намного больший разброс, чем по другим предметам. Это объясняется тем, что усвоение программирования, как и математического материала, требует более интенсивной мыслительной работы, более высокого уровня обобщающей и абстрагирующей деятельности. Этим и объясняется то, что ориентировка на «среднего» ученика в преподавании программирования приводит к низкой успеваемости.

С точки зрения необходимости осуществления индивидуального подхода, очевидно, идеальными условиями обучения являются такие, при которых на одного учителя приходится один обучаемый. Но такие условия нереальны, и поэтому необходимо искать пути осуществления индивидуального подхода при существующей классно-урочной системе организации обучения. В этом направлении можно отметить такие формы обучения, как индивидуальные дополнительные занятия со слабыми студентами (которые должны носить строго

индивидуальный характер) и внеаудиторные занятия с сильными для развития их интереса к программированию.

Индивидуальный подход при обучении программированию обеспечивается оценкой начального стажа и компетенции студента в информатике и программировании, предварительным разделением студентов на относительно однородные группы по уровню начальной компетенции, когда задания на решение задач, также разделены по уровням сложности, и студентам одного уровня компетенции даются одинаковые задания или задания примерно одного уровня сложности.

1.2. Модель процесса развития конструктивно-логического мышления у студентов при изучении программирования

Модель обучения программирования, являясь продолжением и конкретизацией общей теории обучения (дидактики), должна опираться на определенную общедидактическую систему. В качестве общедидактической основы модели обучения программированию нами принята дидактическая система проблемного обучения, согласно которой процесс обучения строится как последовательность проблемных ситуаций [17].

Обучение программированию по своей сути является проблемным обучением, так как является обучением «через задачи».

Обучение через задачи, как в математике, так и в программировании - давно известная и широко обсуждаемая в специальной педагогической литературе проблема. Однако до сих пор она не получила удовлетворительного решения, которое предполагает разработку системы задач, соответствующей современной программе и приспособленной к обучению деятельности по программированию задач для решения на ЭВМ. Это означает, что задачи должны служить и мотивом для дальнейшего развития теории (введения новых понятий, открытия и доказательства новых свойств изучаемых объектов) и возможностью для ее эффективного применения.

В установившейся практике пользуются обучением через задачи лишь при изучении отдельных вопросов. Практика показывает, что такое обучение ведется по схеме «задачи —> теория—> задачи».

Обучение программированию предполагает широкое применение этой схемы. Необходимо исходить из проблемных ситуаций, возникающих вне предметной области программирования (в области естественных наук, техники, экономики или вообще в какой-нибудь области практической деятельности), и формулировки

соответствующих задач. Затем ставится цель решить эти задачи средствами программного моделирования, то есть составления программы для ЭВМ.

Приведем характеристику этапов программирования задачи для решения на ЭВМ и формирование составляющих конструктивно-логического мышления.

Этапы решения задачи на ЭВМ как система мыслительные действий и операций. Первый шаг к достижению цели – решению задачи путем программирования для ЭВМ - состоит в определенной обработке, подготовке задачи, после ее тщательного изучения, в результате чего должна получиться математическая модель фигурирующих в ней сущностей и связей — математическая задача. Возможно, что уже при попытке перевода исходной задачи в математическую или же после осуществления этого перевода, при попытке ее решить окажутся недостаточными имеющиеся в нашем распоряжении математические средства. Эти случаи всегда влияли на историю самой математики (так как стимулировали ее развитие) и так же воздействовали на обучение программированию.

А.А.Столяр подчеркивал, что обучение математической деятельности в сочетании с проблемным обучением не может рассматриваться как некая универсальная методика, применимая во всем процессе обучения математике, на каждом уроке. Сочетание общедидактической системы проблемного обучения с конкретно-методической системой обучения математической деятельности А.А.Столяр представил в виде общей схемы проблемного обучения программированию. Конкретная задача порождает проблемную ситуацию, служит стимулом для расширения теоретических знаний и в конечном итоге для разрешения проблемной ситуации. По мнению А.А.Столяра, его концепция не является универсальной, т.е. применимой ко всем разделам школьного курса математики.

Проблемные ситуации в программировании отличаются от математики тем, что математик создает математические модели, весьма удаленные от реального объекта, с одной стороны, из-за больших упрощений и допущений, а, с другой стороны, он практически не знает ограничения в ресурсах. Программист создает программную модель в жестких рамках ресурсов ЭВМ, в том числе по времени решения задачи, и степень адекватности объекту у такой модели не сравнима с математической – она намного выше.

Схема проблемного обучения программированию, по нашему мнению, должна также существенно отличаться от математической.

В соответствии с технологией и методикой решения задачи на ЭВМ основными показателями конструктивно-логического мышления применительно к программированию являются ЗУН, обеспечивающие следующие действия и объединяемые термином компетенции в программировании:

1. Анализ состава и структуры задачи (задания-проекта), выявление составляющих ее логически законченных подзадач (последующие действия относятся как к подзадачам, так и к задаче, в целом).

2. Преобразование исходной нечеткой вербальной постановки задачи (ПЗ) в формализованную математическую ПЗ.

3. Выбор известного метода или конструирование нового метода решения формализованной задачи.

3. Алгоритмизация метода решения задачи.

4. Выбор структур данных.

5. Выбор языка программирования (ЯП, язык для записи алгоритма, алгоритмический язык, алгоязык) и кодирование алгоритма на ЯП – построение программы.

6. Разработки схемы отладки и отладка программы с использованием тестовых данных.

7. Анализ результатов решения на тестовых данных, сравнение результатов с ручными оценками, и, при необходимости, возврат к пп.1-6.

Наиболее показательными, на наш взгляд, являются пп.1, 3 и 4, для которых могут быть определены уровни КЛМ-АСМ.

Алгоритмизация как ключевой этап решения задачи. Общеизвестно среди педагогов-информатиков, что этап алгоритмизации решения задачи вместе с выбором оптимальной структуры данных является ключевым, так как предопределяет работоспособность и качество итоговой программы. Именно поэтому алгоритмам и алгоритмизации уделяется повышенное внимание в научной, учебной и методической литературе.

Модель обучения. Разработанная экспериментальная система дидактических условий опирается на:

1. Выявленную аналогию между деятельностью по

организационно-экономическому и социальному управлению, с одной стороны, и программированием как управлением задачами, ресурсами и действиями ЭВМ, с другой стороны.

2. Системное соблюдение дидактических принципов:

а) научности и системности (строгое соблюдение технологии решения задачи на ЭВМ; особое внимание этапу составления алгоритма);

в) наглядности (графический анализ структуры задачи, данных и действий, обязательность составления граф-схемы алгоритма ГСА);

г) активности (пошаговая отладка программы с демонстрацией решения на экране монитора ПК).

Обеспечение оптимальных условий для развития конструктивно-логического мышления у студента выступило в виде особой педагогической задачи, в соответствии с которой конкретизировались основные элементы дидактического процесса, а именно:

Формы структурирования содержания учебного материала:

а) *задание-проект* на составление программы как единица содержания;

б) соответствие последовательности изложения учебного материала логике освоения основ алгоритмизации и программирования;

в) соответствие той же последовательности логике развития КЛМ.

Принципы отбора содержания учебного материала:

а) требования к *заданию- проекту*: логическая законченность – от словесной постановки задачи до наглядного пошагового выполнения программы; задания следующего уровня включают элементы предыдущих;

б) различные уровни сложности как в рамках одного задания, так и в рамках тематического раздела.

Методы обучения: использование активных методов обучения:

а) графический метод структурирования задачи, данных и действий;

б) метод демонстрации решения задачи в отладочном режиме;

в) обсуждение со студентом исходного задания, путей и средств решения, результатов, с акцентом на развитие мышления.

Формы организации учебной деятельности студентов: использование активных форм организации учебной деятельности на занятиях: программирование как конструктивно-логическая деятельность по решению задач; ведущая форма - индивидуальная самостоятельная работа и индивидуально-групповая работа на занятиях; когда студенты двух групп работают попарно над общим заданием.

Характер взаимодействия между участниками учебного процесса: партнерские отношения субъектов учебной деятельности: ориентация на субъект-субъектную модель взаимодействия между обучаемыми и обучающим (функция преподавателя состоит в организации успешного обучения); взаимодействие между студентами в паре.

Графическое представление итоговой модели процесса обучения программированию, развивающего конструктивно-логическое мышление с формированием алгоритмического стиля мышления и подготовкой объектному моделированию приведена на рис.1 .

Оценка уровня развития конструктивно-логического мышления через успешность в обучении программирования. Для оценки уровня успешности освоения основ алгоритмизации и программирования, на наш взгляд, можно идти двумя путями:

1) использовать известные тесты или батареи тестов, которые применяются для оценки квалификации профессиональных программистов, например, при их сертификации или приеме на работу;

2) поскольку, на наш взгляд, первый подход неприемлем для оценки ЗУН или компетенции в программировании студентов младших курсов, для большинства из которых не программирование не является избранной областью профессиональной деятельности, то целесообразно разработать тестирующую процедуру специального назначения, которая была бы надежна и валидна, и позволяла достоверно оценивать изменение операциональной составляющей мышления студента, изучающего алгоритмизацию и программирование.

Для проведения измерительных экспериментов по оценке уровня успешности освоения основ алгоритмизации и программирования как уровня развития конструктивно-логического мышления с участием опытных экспертов был разработан измерительный инструментальный, а

именно, тестовая процедура для оценки программистской компетенции и программа (электронная таблица в Excel) обработки и визуализации результатов измерений.

Процедура построена как открытый тест, состоящий из 15 заданий, охватывающих все основные стороны управления данными и действиями при алгоритмизации и программировании задачи. С помощью опытных педагогов-экспертов тестовым заданиям были назначены весовые коэффициенты, соответствующие сложности задания, при этом общая сумма они составила 90 баллов.

Бланк со списком заданий теста приведен в Приложении 3, а их характеристики анализируются ниже.

Для характеристики заданий теста нами используется кортеж из четырех параметров, соответствующих четырем основным действиям по решению задачи на ЭВМ путем программирования:

- «Структурирование» (С – структурирование задачи и ее составляющих, в частности структурирование данных, т.е. ресурсов);
- «Алгоритмизация» (А – алгоритмизация решения задачи и подзадач; структурирование действий и временного ресурса),
- «Программирование» (П – программирование алгоритмов в терминах ЯП; кодирование алгоритма в терминах алгоритмического языка),
- «Отладка» (О – пошаговая проверка сконструированного решения (составленной программы) на тестовых данных),

Сложность каждого аспекта в конкретном тестовом задании задана по трехбалльной шкале (от 1 до 3 – от низкой до высокой). Например, весьма простое задание №1 имеет характеристику 0102, а три наиболее сложных задания (№№ 7, 8, 9) 3333. Такие характеристики использовались автором и экспертами для распределения весовых коэффициентов заданий в тесте.

Таблица 1.1

№№ зада- ний/Хар-ки	С	А	П	О	Балл по заданию
1	0	1	0	2	3
2	1	2	0	2	2
3	0	1	2	1	5
4	3	0	1	1	3*5=15
5	3	1	2	1	6
6	2	3	2	3	6
7	3	3	3	3	7
8	3	3	3	3	7
9	3	3	3	3	8
10	2	2	0	2	8
11	1	1	2	3	2
12	1	1	2	3	4
13	0	0	1	1	4
14	2	3	2	2	7
15	0	2	0	2	6
Итого по характерис- тике	24	30	26	32	
Максимальная сумма баллов по тесту:					90

Уровни успешности в программировании как этапы программы обучения программированию и формирования КЛМ. В соответствии с положениями теорий поэтапного формирования умственных действия и проблемного обучения нами определены пять уровней конструктивно-логического мышления, которым соответствуют уровни успешности в алгоритмизации и программировании задач (параметры уровней – ЗУН алгоритмизации или структурирования действий, структурирования данных, этапы технологии, искусство отладки) (см. табл. выше):

Первый уровень (начальный).

Работа по образцу. Умение составить линейный алгоритм и записать программу последовательной обработки неструктурированных данных без разветвления процесса по условию.

Второй уровень (минимальный).

Работа с модификацией данного образца. Умение составить алгоритм с разветвлениями, записать программу последовательной обработки данных с однократным разветвлением процесса по условию. Использование простейших регулярных структур данных типа одномерной последовательности и регулярных циклов обработки по счетчику.

Третий уровень (хороший, достаточный).

Работа с существенной модификацией данного образца, комбинирование из различных образцов. Умение записывать программу последовательной обработки данных с многократным разветвлением процесса по условию, с вложением разветвлений. Использование различных регулярных структур данных (одномерных и многомерных последовательностей, векторов и матриц, записей и файлов) и всех регулярных циклов обработки - по счетчику, по предусловию, по постусловию.

Четвертый уровень (продвинутый).

Работа без опоры на образец. Умение составить алгоритм высокой сложности и записать программу обработки данных с многократным разветвлением процесса по условию, с вложением разветвлений. Использование различных регулярных структур данных и всех регулярных циклов обработки - по счетчику, по предусловию, по постусловию, с умением преобразовать один тип цикла в другой. Умение реализовать

циклический процесс через рекурсивные процедуры. Работа с динамической памятью и динамическими структурами данных.

Пятый уровень (высокий).

Возможен переход в другую парадигму программирования, например, переход от процедурно-ориентированного к объектно-ориентированному программированию.

Дидактические условия развития конструктивно-логического мышления студентов при обучении программированию. Деятельностный подход к развивающему обучению программированию означает обучение студентов через решение ими задач нарастающей сложности на компьютере путем программирования на некотором доступном алгоритмическом языке, то есть через моделирование деятельности программиста.

Проанализируем два варианта дидактических условий (ДУ) и оценим их развивающий потенциал:

1. Система ДУ, применявшаяся до внедрения действующих учебных стандартов (до 2000 года). В рамках отведенных учебных часов, помимо лекций и практических работ с использованием компьютера, выделялись часы для семинаров, на которых без ЭВМ (без ПК – персонального компьютера), в групповой форме детально разбирались начальные этапы программирования задачи, прорисовывались структурные схемы и граф-схемы алгоритмов. Отсутствие в тот период у большинства студентов личных ПК (не считая периодических коротких занятий-практикумов с участием преподавателя) оставляло для них только одну возможность практической работы, а именно, самостоятельную работу в вечернее время в компьютерных классах вуза, свободных от плановых занятий.

Определенные возможности обучающихся для практикума в часы самоподготовки в компьютерном классе дополнялись глубокой фундаментальной проработкой заданий в аудитории.

2. Действующая система ДУ.

После введения в 2000 году действующих стандартов высшего профессионального образования, в которых сделан акцент на самостоятельную работу студента, общее число часов, выделенных на изучение основ алгоритмизации и программирования, заметно сократилось, при этом семинарские занятия из планов полностью исключены, то есть работа совместная учебная деятельность

преподавателя со студентом в аудитории, в отсутствие компьютера, практически отсутствует. Акцент на самостоятельную работу студента при наличии у большинства студентов личных ПК, привели к тому, что неорганизованная самостоятельная работа студента стала проходить вне вуза, т.е. стала внеаудиторной, «домашней» - без контроля и участия преподавателя.

Результатом такой перестройки, на наш взгляд, стало заметное снижение результативности обучения программированию.

Таким образом, с учетом реальностей существующей ситуации и действующих учебных планов, дидактические условия развивающего обучения программированию, по нашему мнению, должны включать в себя лучшее из компонентов ДУ двух систем – например, фундаментальную проработку заданий и их решений, характерную для первой системы, а также возросшие возможности студентов по самостоятельной работе с персональным компьютером, нейтрализуя их недостатки, такие как расширение самостоятельности при отсутствии специальной организованности обучающихся с одновременным снижением системности и научности в их обучении.

1.3. Программа развития конструктивно-логического мышления у студентов при изучении основ программирования

Действующими учебными планами специальностей и рабочими программами курсов информатики и основ программирования предусмотрены две формы занятий – лекции и практические занятия в компьютерной лаборатории.

Лекции одного содержания проводятся совместно для всех групп, а для практических занятий академическая группа делится на подгруппы по числу рабочих мест в компьютерном классе. Общий лекционный материал является единым содержательным компонентом для всех студентов. Содержание практических занятий также едино, но операционный компонент существенно различается для двух категорий студентов-участников эксперимента.

Содержанием экспериментальной программы развивающего обучения программированию заключались в следующем:

- выявить этапы развития конструктивно-логического мышления при обучении программированию и определить соответственно к ним задания и задачи, критерии развития конструктивно-логического мышления на том или ином этапе;

- научить каждого учащегося, независимо от исходного уровня сформированности конструктивно-логического мышления, умению ставить и решать задачи путем их программирования для ЭВМ, поднять каждого студента до среднего уровня компетентности;

- показать студенту, как пользоваться знаниями, умениями и навыками, полученными на занятиях по программированию в решении профессиональных задач его специальности. Дать теоретические основы и приемы развития КЛМ на уроках информатики студентам - будущим преподавателям математики и информатики.

Таким образом, образовательные цели формирующего эксперимента предполагали создание дидактических условий для поэтапного развития конструктивно-логического мышления студентов.

Для этого в эксперимент для экспериментальной группы студентов были добавлены следующие задания, которые выполнялись строго графически на бумаге:

1) анализ структуры задачи и ее прорисовка в виде дерева подзадач;

2) анализ структуры данных и их спецификация;

3) построение ГСА всех ключевых подзадач и задачи в целом;

4) пошаговая отладка синтаксически правильной программы с заданием контрольных точек останова и контролем переменных программы.

5) письменный и устный отчет по проделанной дополнительной работе.

На наш взгляд, в этих заданиях воплощались принципы научности, системности, наглядности.

На основе созданной модели была разработана экспериментальная программа развивающего обучения программированию, реализующая выявленные дидактические условия и рассчитанная на два семестра. Задания-проекты, выполняемые студентами, реализуют концепцию проблемного обучения (задания нарастают по сложности, задания следующего уровня включают содержательные и операционные элементы предыдущего уровня) и направлены на поэтапное развитие конструктивно-логического мышления студента. В первом семестре студенты осваивают основы структурного анализа задач и их решения, составления простых и средней сложности алгоритмов, с

реализацией на языке программирования Турбо-Паскаль. Сложность заданий нарастает от линейных вычислений сложных выражений до использования однородных структур данных и действий (линейные последовательности, одномерные массивы, циклы, файлы простой структуры, статичная графика). Оценка достигнутого уровня успешности в программировании (и уровня развития конструктивно-логического мышления в сочетании с алгоритмическим стилем мышления) в конце первого семестра позволяет оценить зону ближайшего развития каждого студента, составить прогноз и уточнить план дальнейшей работы с конкретным студентом. Программа второго семестра включает решение задач первого семестра, но в усложненной постановке, с использованием других структурных решений и средств алгоритмического языка (рекурсия, записи, структурированные файлы, динамические структуры, анимированная графика, объектно-ориентированное программирование). Студенты, в разное время достигая своего предела в развитии исследуемого вида мышления, продолжают работать по образцу и закреплять ЗУН, приобретенные на ранних этапах программы. Отдельные студенты, потенциал развития которых не был исчерпан и в процессе обучения начал раскрываться, в освоении программы проходят все этапы и в конце экспериментального периода составляют программы, соответствующие начальному уровню и стилю профессионального программирования. Они продолжают обучение по программе до знакомства с основами объектно-ориентированного программирования в рамках среды Object Pascal.

Экспериментальная программа рассчитана на учебный год или на два семестра занятий, что соответствует стандарту подготовки по основам информатики и программирования у студентов естественнонаучных специальностей. Программа состоит из пятнадцати заданий-проектов (см. Приложение 2) и разбита на два этапа, то есть этап обучения это семестр. Программа представлена в виде таблицы и в ней этап обучения программированию соответствует этапу развития конструктивно-логического мышления с формированием алгоритмического стиля мышления.

Программа развивающего обучения основам алгоритмизации и программирования приведена в Приложении, а фрагмент программы приводится ниже в третьей главе.

Цели первого этапа - «Введение в алгоритмизацию и программирование» (1-ый семестр занятий):

- сформировать у студента умения и навыки программирования с неструктурированными однородными данными и действиями (см. «Характер контрольного среза» в Прил.2), что в плане развития КЛМ-АСМ соответствует умению выполнять простые однородные операции над ресурсами и действиями для решения задачи. Развиваемые на этапе способности, умения и навыки перечислены в колонке «Характер контрольного среза», а варианты заданий приведены в колонке «Содержание рассматриваемых вопросов» Приложения 2;

- сформировать у студента умения и навыки организовать программную обработку регулярных структур данных с использованием регулярных структур действий, что в плане развития КЛМ-АСМ соответствует регулярному манипулированию и простому структурированию регулярных ресурсов решаемой задачи. Развиваемые на этапе способности, умения и навыки перечислены в колонке «Характер контрольного среза» Прил. 2, а варианты заданий приведены в колонке «Содержание рассматриваемых вопросов» с номерами из таблицы в Приложения 2.

Цели второго этапа - «Изучение основ алгоритмизации и программирования» (2-ой семестр занятий):

- сформировать у студента умения и навыки программного управления многоуровневыми и вложенными структурами данных и действий, типами данных, что в плане развития КЛМ-АСМ соответствует умению строить и использовать структурирование и иерархию ресурсов и действий с выделением агрегатов. Развиваемые на этапе способности, умения и навыки перечислены в колонке «Характер контрольного среза», а варианты заданий приведены в колонке «Содержание рассматриваемых вопросов» с номерами из таблицы в Приложении 2;

- начать освоение студентом, достигшим продвинутого уровня успешности в алгоритмическом программировании и соответствующего уровня развития КЛМ-АСМ, другой парадигмы программного управления данными и действиями - объектной, которая соответствует умению увидеть ресурсы и действия по решению задачи как единое целое – агрегат, объект, соответствующий объекту или явлению изучаемой предметной области. Отсюда задачей четвертого этапа является разбор совместно с преподавателем

конкретных примеров моделирования для предметных областей, самостоятельная практическая реализация студентом отдельных фрагментов задачи, программируемых в объектно-ориентированной среде.

Выводы

На основе личного педагогического опыта и по результатам анализа литературы по психологии и педагогике программирования выявлено содержание деятельности программиста, а также деятельности студента, изучающего программирование [48]. Выявлены операциональный состав этого вида деятельности, этапы и стадии ее освоения. Проведен анализ и дана характеристика этапов решения задачи на ЭВМ с позиций формирования составляющих конструктивно-логического мышления. Этапы структурирования задачи и алгоритмизации выделены как ключевые.

Анализ научно-педагогической литературы по проблеме развития практического мышления, а также дидактических исследований по алгоритмизации и программированию, личный педагогический опыт автора позволили выбрать в качестве базового подхода для развивающего обучения программированию студентов использование: а) структурирования с визуализацией задачи, данных и действий; б) структурного и процедурного программирования; в) пошаговой отладки программы с визуализацией всех компонентов решения.

Развитие КЛМ – динамический процесс. Предложенная модель процесса развития КЛМ у студента последовательный переход от начального уровня компетенции в программировании, определяемого исходным уровнем подготовки студента как выпускника среднего учебного заведения, его стажем в информатике и программировании, к максимально возможному уровню для конкретного студента, определяемому его способностями. Переход характеризуется этапностью и соответствием, как логике освоения основ алгоритмизации и программирования, так и логике развития КЛМ.

Выявлено, что необходимыми дидактическими условиями развития КЛМ у студента, изучающего основы алгоритмизации и программирования, являются:

- а) *формы структурирования содержания учебного материала:*

- *задание-проект* на составление программы как единица содержания;

- соответствие последовательности изложения учебного материала логике освоения основ алгоритмизации и программирования;

- соответствие той же последовательности логике развития КЛМ.

б) принципы отбора содержания учебного материала:

- требования к *заданию-проекту*: логическая законченность — от словесной постановки задачи до наглядного пошагового выполнения программы; задания следующего уровня включают элементы предыдущих;

- различные уровни сложности как в рамках одного задания, так и в рамках тематического раздела.

в) методы обучения:

- использование активных методов обучения: графический метод структурирования задачи, данных и действий;

- метод демонстрации решения задачи в отладочном режиме;

- обсуждение со студентом исходного задания, путей и средств решения, результатов, с акцентом на развитие мышления.

г) формы организации учебной деятельности студентов: использование активных форм организации учебной деятельности на занятиях: программирование как конструктивно-логическая деятельность по решению задач; ведущая форма - индивидуальная самостоятельная работа, а также индивидуально-групповая работа на занятиях, когда студенты двух групп работают попарно над общим заданием.

д) характер взаимодействия между участниками учебного процесса: партнерские отношения субъектов учебной деятельности: ориентация на субъект-субъектную модель взаимодействия между обучаемыми и обучающим (функция преподавателя состоит в организации успешного обучения); взаимодействие между студентами в паре.

Разработана модель развивающего обучения программированию, включающая выявленные дидактические условия. На основе модели разработана экспериментальная программа развивающего обучения студентов основам алгоритмизации и программирования.

Глава 2. Опытнo-экспериментальная работа по обучению программированию и развитию конструктивно-логического мышления студентов

В настоящей главе описываются три этапа педагогического эксперимента – констатирующий, формирующий и контрольный, анализируются и обобщаются их результаты. Основной задачей экспериментального исследования было, с одной стороны, показать эффективность разработанных дидактических условий, а, с другой стороны, доказать, что разработанная на основе предложенной модели методика обучения способствует повышению успешности обучения программированию, и тем самым обеспечивает развитие конструктивно-логического мышления студентов.

Экспериментальная работа проводилась на протяжении нескольких лет и была разбита на три этапа – констатирующий этап, формирующий эксперимент по обучению и контрольный этап.

Констатирующий этап (1998-2003 гг.) включал в себя:

1) отработку методических и организационных элементов формирующего эксперимента;

2) проверку разработанной диагностической процедуры оценки успешности в программировании, проверку ее валидности и надежности;

3) проверку взаимозависимости программистской компетенции, уровня общего развития мышления и уровня развития задатков КЛМ-АСМ в раннем дошкольном, среднем школьном возрасте и на первых курсах вуза.

На протяжении нескольких лет проводилось исследование развития конструктивно-логического мышления у студентов-математиков 1-го и 2-го курсов через обучение алгоритмизации и программированию. Проверялась связь между развитием задатков конструктивно-логического мышления и алгоритмического стиля мышления у обучающихся в ключевые возрастные периоды - а именно, в раннем дошкольном возрасте (проводилось анкетирование родителей студентов по зарождению конструктивной деятельности у ребенка в раннем дошкольном возрасте), в средних и старших классах школы (проводился опрос студентов по срокам и продолжительности опыта компьютерного дела, изучения ОИВТ и программирования), когда начинается систематическое изучение основ математики и одновременно у

части подростков происходит первое знакомство с компьютерами и программированием. Если в указанные сенситивные периоды ребенку были созданы условия для развития мышления, то, в первом случае, одновременно зарождаются конструктивная деятельность и логическое мышление ребенка, а, во втором случае - на уроках математики или на дополнительных занятиях в кружке - математика и программирование наиболее естественным образом укладываются в голове школьника, тогда образуя когнитивный психоконструкс КЛМ-АСМ, тогда, к окончанию школы, результаты развития ребенка особенно значительны. Такие старшеклассники составляют контингент математических олимпиад и олимпиад по программированию всех уровней – от районного до международного.

2.1. Организация и методика констатирующего этапа экспериментального исследования

Констатирующий эксперимент включал:

- отработку организационных моментов для обучающего эксперимента;

- отработку процедуры измерения успешности в изучении алгоритмизации и программирования и методики применения процедуры.

В предыдущих разделах работы обоснована актуальность задачи исследования различных психологических и педагогических аспектов успешности, как программирования, так и обучения программированию, выявления предпосылок к освоению этой профессии, определения способностей конкретного студента, и, наконец, построения процесса обучения в соответствии с групповыми и индивидуальными особенностями обучаемых.

Решались следующие задачи констатирующего этапа исследования:

- изучалась литература по теме исследования;

- формулировались цели, задачи и гипотезы исследования;

- выбирались и разрабатывались психодиагностические методики для определения общих познавательных способностей и специальных конструктивных способностей;

- проводилось психодиагностическое исследование заданной выборки студентов;

- проводилась статистическая обработка результатов диагностики, выявлялись показатели успешности обучения и связи между отдельными показателями;

- интерпретировались результаты обработки и делались обоснованные выводы.

Результаты этапа легли в основу модели развивающего обучения программированию и экспериментальной программы обучения для формирующего этапа.

Ниже кратко описывается и анализируется процесс подготовки в области программирования студентов первого курса, обучающихся на трех специальностях математического факультета УдГУ: информатики (спец. "Информационные системы"), математики-системные программисты (спец. "Прикладная математика и информатика") и математики-будущие преподаватели математики и информатики (спец. "Математика").

Объект этапа исследования

В первом этапе исследования приняли участие студенты 1-го курса математического факультета УдГУ, т.е. обследуемая выборка характеризуется возрастной однородностью (17-18 лет) и имела следующую структуру:

- всего участвовало 86 человек из 100, зачисленных на 1-й курс МФ летом 1998 года (часть студентов прекратила обучение по разным причинам или отказались от участия в исследовании);
- выборка поделена на 4 академические группы численностью

Таблица 2.1

Выборки студентов для констатирующего эксперимента

Категория\Группа	М	М	ПМИ	ИС	Всего
Юноши	6	5	6	19	36
Девушки	19	14	9	8	50
Городские	17	11	14	21	53
Сельские	7	8	2	6	23
Вып-ки лицеев, гимназий	6	3	8	12	29
Всего	25	19	15	27	86

В то же время вся выборка естественным образом делится примерно пополам, т.к. две первые группы - это будущие математики-педагоги (не программисты, 44 чел.), две последующие - будущие системные программисты и информатики (42 чел.), объединяемые термином "программисты".

Предмет констатирующего исследования:

- программирование как вид конструктивной деятельности и его структура;
- общие познавательные способности студентов-математиков, их задатки и специальные конструктивные способности как факторы успешности обучения программированию;
- половые различия в обучении и практике программирования;
- методические основы оценки способностей и определения прогноза успешности освоения студентами профессиональных знаний, умений и навыков программиста;
- организация и методика обучения студентов 1-го курса началам программирования.

Гипотезы, проверяемые в констатирующем эксперименте:

- 1) способности к программированию не связаны непосредственно с общими познавательными способностями, в целом, как не связаны и с математическими способностями;
- 2) способности к конструктивной деятельности, в целом, и к программированию, в частности, в значительной мере формируются на основе задатков в дошкольном возрасте и младшем школьном возрасте, но получают свое развитие до уровня способностей к программированию, в сочетании со склонностями к профессии программиста, в средних и старших классах школы, и только при создании соответствующих условий;
- 3) система обучения по дисциплине "Основы информатики и вычислительной техники" в средней школе при должной организации и уровне оснащения, которые характерны для городских лицеев и гимназий, создает решающие предпосылки для успешной профессиональной подготовки по программированию.

Организация и методики констатирующего экспериментального исследования.

Исследование проводилось в четыре подэтапа. Тестирование по отдельным методикам проводилось с интервалом в 2-3 недели, либо

во время занятий, либо после занятий. Большинство студентов проявило заинтересованность в исследовании своих способностей и понимание важности проводимых исследований.

На первом подэтапе все участвующие в работе студенты прошли тестирование познавательных способностей с помощью тестов КОТ (краткий ориентационный тест), "Прогрессивные матрицы Равена", а также 6-го, 7-го и 8-го субтестов Амтхауэра.

Выбор тестов обусловлен следующими соображениями.

1. *Краткий ориентационный тест (КОТ)*, содержащий 50 разнотипных заданий, был проведен первым и, на наш взгляд, помог исследователям и студентам подготовиться к проведению более сложных сертифицированных тестов, и одновременно позволил предварительно ранжировать студентов по уровню развития познавательных способностей. Использован бланковый вариант теста. Разработана программа-электронная таблица в MS Excel и методика оценки результатов.

2. *Методика на наглядно-образное мышление*. Использован тест И.К.Равена (1936 г.). На успешность его выполнения влияют уровень развития внимания, восприятия и образного мышления. В нашем исследовании был применен полный вариант методики, состоящей из 60 картинок. Тест состоял из 5 серий по 12 картинок:

- 1) Установление взаимосвязи в структуре матрицы.
- 2) Отыскание аналогии между парами фигур.
- 3) Отыскание прогрессивных изменений в фигурах матрицы;
- 4) Нахождение принципа перегруппировки фигур по горизонтали.
- 5) Нахождение принципа анализа и синтеза фигур на элементы.

Фиксируется время выполнения теста, ошибки, время выполнения отдельных заданий, выводится модифицированный балл. Использован бланковый вариант теста. Разработана программа в MS Excel и методика оценки результатов.

3. *Методика на числовое логическое мышление*. Мысленное оперирование числами отлично от операций с понятиями. поэтому для его исследования был выбран "тест числового ряда" - субтест 6 теста Амтхауэра. Числовые ряды, предлагаемые испытуемому, построены по определенной закономерности. каждом из них не хватает завершающего числа. Испытуемый должен раскрыть закономерность и закончить последовательность. Общее число рядов – 20; время работы - 10 мин. Фиксируется время выполнения

задания и ошибки; "сырые" данные переводятся в балльные оценки. Использован бланковый вариант теста. Разработана программа в MS Excel и методика оценки результатов.

4. *Методики на пространственные представления*. Использовались субтесты № 7, 8 теста Амтхауэра. Первый состоял из 16 заданий. Испытуемому предлагались картинки с изображенными на них разрезанными плоскими фигурами. Требовалось идентифицировать каждую разрезанную фигуру с одним из образцов.

Второй подэтап первого этапа состоял из 20 заданий. Испытуемому предлагались картинки с изображенными двумя рядами кубов. Верхний ряд - 5 кубов, повернутых так, что видно только 3 из шести граней. Нижний ряд - другие кубы, среди которых нужно найти один из верхних, но ориентированный по-другому. фиксировалось время и число правильных ответов. Использован бланковый вариант теста. Разработана программа в MS Excel и методика оценки результатов.

Обобщенные результаты тестирования приведены в таблицах.

На втором подэтапе, проводимом параллельно с первым, нами совместно с экспертами кафедр общей психологии и педагогики и педагогической психологии УдГУ была разработана анкета для родителей (см. Приложение 2). В ней автор, обращаясь к родителям (в первую очередь к матери) конкретного студента, просил ответить на ряд вопросов и дать, по возможности, полные сведения о тех видах игровой и познавательной деятельности, в которых ребенок проявлял активность в дошкольном и школьном возрасте. На наш взгляд, ответы на поставленные вопросы, позволяют, в определенной мере, установить направленность и выраженность природных задатков к конструктивной деятельности у будущего студента, а также направленность и активность его учебной и познавательной деятельности в дошкольном и школьном возрасте, формирование способностей и склонностей.

Первый раздел относится к посещению ребенком детских дошкольных учреждений, что позволяет определить уровень его вовлеченности в учебно-воспитательную деятельность по унифицированным программам. Количественно этот фактор оценивался двумя отрезками (от 2-5 и от 5-7 лет), которым были назначены разные весовые коэффициенты.

Вопрос второго раздела получился малоинформативным и на него был получен практически стандартный ответ.

В третьем разделе родителям предлагалось отметить те виды игрушек, которые чаще всего использовал ребенок в возрасте до трех лет. Здесь явно перечислены 11 видов игрушек, а в 12-м пункте можно было вписать дополнительные виды, не вошедшие в список. В общую сумму пункты этого раздела включались со следующими весовыми коэффициентами, которые, по нашему мнению, отражают их вклад в формирование у ребенка задатков конструктивной деятельности (см. табл. X).

Наличие отметки некоторого вида игрушки в анкете давало в итоговую сумму дополнительное слагаемое, равное назначенному весовому коэффициенту этого вида.

В 4-ом разделе выяснялись игры и занятия, которые преобладали в деятельности ребенка до 7 лет. В общую сумму пункты этого раздела включались со следующими весовыми коэффициентами, которые, по нашему мнению, отражают их вклад в формирование у ребенка задатков конструктивной деятельности (см. табл. 5).

В 5-ом разделе оценивалась способность ребенка определенное время заниматься любимым делом, характеризующая произвольность его познавательных процессов: определен диапазон от 5 мин, через 10, 20, 30 минут до периода, названного родителями. Здесь нами использована нелинейная шкала весов:

- До 5 мин	= 1
- До 10 мин	= 3
- До 20 мин	= 5
- До 30 мин	= 7
- Более 30 мин	= 9

В 6-м и 7-м разделах собирались сведения о кружках и секциях, которые посещал ребенок в школьные годы, какие увлечения он имел продолжительное время. Здесь производилось простой подсчет количества названных пунктов.

Была разработана методика обработки, целями которой были выявление и количественная оценка конструктивной стороны игровой и учебной деятельности ребенка, которые проявляли соответствующие задатки и способствовала развитию у него способностей, лежащих в основе будущей деятельности по программированию. Основным элементом методики являются шкалы весов, которые приведены в таблицах 5 и 6.

На этом этапе также был проведен опрос всех студентов на предмет их стажа в программировании до поступления в вуз. При этом фиксировались: общее число лет занятий программированием, из них выделялись периоды начального освоения алгоритмизации и программирования, а также программирование на языке Бейсик (этот стаж засчитывался с коэффициентом 1), и отдельно - период программирования на языках профессионального уровня с развитыми механизмами структурирования данных и действий, таких как Паскаль, Си или Ассемблер (этот стаж засчитывался с коэффициентом 2). Период освоения пользовательского компьютерного курса не засчитывался. В итоге для каждого студента был получен числовой код - его условный стаж программирования (см. графу "Стаж" в табл. 2.5).

На третьем подэтапе с участием экспертов-преподавателей информатики математического факультета был разработан специальный тест по программированию (Приложение 3). Тест - открытого типа, для него методом экспертных оценок были введены оценочные баллы для каждого из 15 заданий, максимальное число баллов по тесту - 90. Время выполнения теста не фиксировалось, но эмпирически было выявлено, что для профессионального программиста на его выполнение требуется 20-30 минут, тогда как студентам-первокурсникам на его выполнение отводилось два академических часа. Первый опыт применения теста показал, что:

а) это перспективное направление построения объективных измерительных процедур различной степени сложности для оценки уровня ЗУН по программированию;

б) эти процедуры хорошо формализованы и естественным образом переносятся на компьютер;

Таблица 2.2

Весовые коэффициенты видов игровой деятельности

Вид игрушек	Весовой коэф-т
1. Кубики	3
2. Мозаика	2
3. Куклы, мягкие игрушки	0
4. Машинки	0
5. Солдатики	0

6. Разборные игрушки	3
7. Строительные наборы	2
8. Посуда, формочки	1
9. Карандаши, палочки	1
10. Книжки	0
11. Музыкальные игрушки, в т.ч. клавишные	2
12. Что-либо еще	-

Таблица 2.3
Весовые коэффициенты видов конструктивной деятельности

Вид игр или занятий	Весовой коэф-т
1. Лепка	3
2. Рисование	2
3. Раскрашивание	1
4. Срисовывание, другая работа по образцу	1
5. Складывание мозаики; собирание кубиков	2
6. Разрезные картинки	3
7. Строительные наборы	3
8. Изучение устройства бытовых предметов	3
9. Создание одежды для кукол и игрушек	2
10. Использование предметов и игрушек новым способом	1
11. Подражание хозяйственной работе взрослых	2
12. Конструирование из готовых деталей	1
13. Конструирование из подручных материалов	3
14. Игра на музыкальных инструментах	2
15. Что-либо еще	-

в) тест получился достаточно сложным - лишь единицы из наиболее продвинутых студентов сумели выполнить некоторые из заданий и за счет этого набрать 70-80 баллов, в то время как часть

студентов (в первую очередь девушки и студенты платной формы обучения из группы ИС) вообще не смогли выполнить тест до конца из-за его высокой сложности.

На четвертом этапе проводилась статистическая обработка результатов тестов общих познавательных способностей и специального теста, данных анкет.

Обработка и интерпретация результатов констатирующего эксперимента.

Результаты статистической обработки приведены ниже в таблицах.

Для всех выделенных категорий студентов (Математики-Программисты, Девушки-Юноши, Городские-Сельские, Лицеисты-Городские Школьники- Сельские Школьники) подсчитывались средние показатели (X), показатели, характеризующие разброс значений, проводилось сравнение для средних при помощи t-критерия Стьюдента при двух уровнях значимости, построены таблицы интеркорреляций для Математиков и Программистов.

Таблица 2.4
Оценки общих познавательных способностей

Группа	КОТ	Равен	Амт6	Амт7	Амт8	Амт	Сумма
Мат-1	22,6	51,2	11,4	11,9	11,5	37,1	110,8
Мат-2	24,9	49,1	12,5	12,1	11,9	38,0	112,1
Прог-1	22,3	50,5	11,9	12,3	11,1	35,7	108,5
Прог-2	24,5	50,5	11,0	12,4	11,6	35,0	109,6

Средние показатели скрывают значительный разброс результатов между крайними значениями. Так, лучшие показатели отдельных студентов по суммарному показателю общих познавательных способностей составили 130-135 баллов, что в сравнении с худшими показателями (80-85), дает превышение на 60%. Такова разница между наиболее сильными выпускниками городских лицеев и отдельными выпускниками сельских школ, а

также студентами, обучающимися на платной основе. При этом зафиксированы два абсолютных показателя (20 баллов из 20) в субтестах Амтхауера у студентов-девушек, одна из которых закончила городской лицей № 41 г. Ижевска, а другая - выпускник средней школы из райцентра Удмуртии.

Обращает на себя внимание колонка с показателями по методике Равена, которые характеризуют уровень развития интеллекта у подавляющего большинства студентов МФ как «выше среднего».

Результаты специального теста (средние показатели) приведены в следующей таблице вместе со средними показателями баллов по анкете и стажу программирования.

Таблица 2.5

Результаты оценки специальных способностей

Группа	Анкета	Стаж	Спецтест
11-11	34.2	2.52	35.87
11-13	31.8	2.26	43.5
19-11	44.4	5.38	52.9
39-11	54.6	5.09	51.7
Математики	32.3	2.43	39.1
Программисты	46.0	5.19	52.8

* для уровня достоверности $p > 0,05$;

** для уровня достоверности $p > 0,01$.

Из таблицы видно, что показатели анкеты у программистов несколько выше, чем у математиков (уровень достоверности различий по t-тесту Стьюдента 0,01), тогда как показатель стажа больше в 2-2,5 раза, давая существенное возрастание показателя по спецтесту. Превышение последнего у группы ИС был бы еще значительнее, но его заметно снизили чрезвычайно низкие показатели девушек из группы ИС платной формы обучения (14 и 16 баллов), которые выглядят удручающе рядом с типичными показателями для этих групп (55-65). Лучшими по спецтесту оказались показатели троих "программистов": девушка из группы

19-11 (78 баллов) и двое юношей из группы ИС (78 и 80 баллов из 90 возможных).

Абсолютно лучший итоговый результат показал студент группы ИС Иван Ч. (135 баллов по тестам познавательных способностей + 70 баллов по спецтесту = 205), показатель которого по анкете выглядит весьма скромно (13). В то же время, в результате беседы стало известно, что он вырос в семье профессионального программиста и приобщаться к компьютерному делу, в целом, и к программированию, в частности, начал с 5-го класса, т.е. на рубеже начальной и средней школы. То есть имела место ранняя ориентация на профессию и ярко выраженная склонность к занятиям программированием, т.е. явно присутствующие задатки получили своевременную стимуляцию.

Интересный результат показала и студентка Ирина К. из группы прикладных математиков: результаты тестов познавательных способностей – $30+48+11+14+14 (=117)$ + спецтест (78) = 195. Результат обработки ее анкеты равен 34 – это самый высокий показатель для всей выборки. В беседе со студенткой установлено, что ее мать работает программистом, но приобщила свою дочь к программированию только год назад, т.е. в выпускном классе школы. Тем самым, явно выраженные задатки, развитые в конструктивные способности, и даже поздний профессиональный выбор, не помешал развитым конструктивным способностям проявиться и в программировании.

Анализ результатов констатирующего эксперимента позволяет сделать следующие выводы:

1) подавляющее большинство студентов-первокурсников МФ имеет уровень развития познавательных способностей высокий или очень высокий, что объясняется высокими требованиями при приеме на факультет.

2) Из таблицы 2.4 видно, что показатели развития общих познавательных способностей практически одинаковы для всех четырех групп. Это подтверждает известное мнение о том, что на естественнонаучные специальности поступают выпускники, сориентированные на них еще в школе и обучающиеся, как правило, в специализированных школах или классах, или готовятся самостоятельно, или под руководством репетитора, но, в любом случае, фактор случайности очень незначителен. Случаи же прекращения занятий до первой сессии или отсева после нее произошли до начала исследования (14 человек из 100).

3) Прогноз успешности дальнейшего обучения программированию студентов, вошедших в исследуемую выборку, предлагается строить на основе результатов, которые они показали по специальному тесту.

Эти данные (Табл. 2.6) показывают, что все студенты делятся на три неравные группы – высокой успешности (Высокая – 50-80), средней успешности (Средняя – 30-50) и низкой успешности в обучении программированию (Низкая – до 30).

Из таблицы видно, что у математиков по четверти студентов показали высокий и низкий результаты, а ровно половина студентов – девушки и юноши, показали среднюю успешность. Тогда как у программистов больше половины девушек показали высокий результат и по четверти – средний и низкий, и 88% юношей – высокий, и только 1 студент – низкий. Т.е. показатели программистов в 2,5 раза превосходят аналогичные показатели математиков, показатели юношей почти в 2 раза выше, чем у девушек. Из таблицы видна значительная поляризация результатов у юношей факультета, но не у программистов, где собрались рано сориентированные на исследуемый вид деятельности.

4) интеркорреляционный анализ, результаты которого приведены ниже в таблицах 2.7 и 2.8, показал наличие сильных парных зависимостей между следующими показателями групп (специальностей). Все значения в таблице определены на уровне достоверности $p < 0,01$.

5) Из таблиц видно, что результаты по спецтесту у математиков не коррелируют с какими-либо показателями тестов общих способностей, в то время как у программистов эти результаты не только существенно выше, но и на высоком уровне значимости (коэфф. корр. = 0,643 при уровне достоверности 0,01) зависит от стажа студента в области программирования.

6) Тесты общих способностей показали, что рост численности студентов-девушек существенно повышает средние показатели группы; но эти показатели снижаются с ростом числа студентов-горожан и числа выпускников лицеев. Это может быть объяснено тем, что в УдГУ поступают безусловно лучшие выпускники сельских школ (в нашей выборке большое число сельских выпускников-медалистов), для которых характерно полноценное разностороннее развитие. Уровень подготовки в периферийных городских школах в последние годы неуклонно снижается. В то же время, лицейская подготовка нередко носит узкоспециальный

характер, особенно для старшеклассников, проявивших способности в отдельных предметах, для них учеба в старших классах нередко заменяется тренировками по одному предмету и выступлениями на всевозможных предметных олимпиадах.

7) Именно названные выше причины объясняют прямо противоположную картину по результатам специального теста:

Таблица 2.6

Распределение студентов по группам успешности

Группа	Высокая (%)	Средняя (%)	Низкая (%)
Мат-1	16	48	36
Девушки	26	42	32
Юноши	17	50	33
Мат-2	26	42	32
Девушки	29	57	14
Юноши	40	60	-
Математики	27	50	23
Девушки	27	48	25
Юноши	27	54	19
Прог-1	80	16	4
Девушки	77	23	-
Юноши	75	13	12
Прог-2	63	11	26
Девушки	25	25	50
Юноши	95	5	-
Программисты	74	14	12
Девушки	52	24	24
Юноши	88	8	4
Все студенты	50	33	17
Все девушки	36	40	24
Все юноши	69	22	9

Таблицы 2.7-2.8

Матрица интеркорреляций для категории "Математики"

	КОТ	Равен	Амт6	Амт7	Амт8	Амт	Анкета	Стаж	Спец	Общ
КОТ	X	0,269	0,202	0,323	0,309	0,357	0,091	0,060	0,399	0,699
Равен		X	0,382	0,165	0,084	0,264	-0,057	0,159	0,031	0,605
Амт6			X	0,343	0,439	0,683	0,136	-0,098	0,386	0,630
Амт7				X	0,413	0,496	0,127	0,360	-0,023	0,488
Амт8					X	0,738	0,096	-0,009	0,313	0,606
Амт						X	0,117	-0,063	0,345	0,847
Анкет							X	-0,057	0,245	0,086
Стаж								X	-0,188	0,038
Спец									X	0,371
Общ										X

Матрица интеркорреляций для категории "Программисты"

	КОТ	Равен	Амт6	Амт7	Амт8	Амт	Анкета	Стаж	Спец	Общ
КОТ	X	0,208	0,279	0,068	0,218	0,254	0,451	0,508	0,531	0,628
Равен		X	0,215	0,233	0,145	0,296	0,374	0,278	0,512	0,642
Амт6			X	-	0,006	0,375	0,051	0,306	0,383	0,422
Амт7				X	0,530	0,809	0,316	-0,146	0,083	0,614
Амт8					X	0,802	0,423	-0,079	0,152	0,635
Амт						X	0,372	0,004	0,272	0,828
Анкета							X	0,125	0,206	0,521
Стаж								X	0,643	0,387
Спец									X	0,567
Общ										X

8) Именно названные выше причины объясняют прямо противоположную картину по результатам специального теста:

Таблица 2.9

Результаты измерения компетенции в программировании

	Дев	Юн	Гор	Сель	Лиц	ГорШ	СелШ
Математики	38.4	50.0	42.8	44.3	36.8	44.9	44.8
Програм-ты	45.1	59.3	57.7	34.8	65.0	38.9	40.0
Все студенты	40.9	52.6	46.8	41.9	56.4	40.7	43.4

* Все различия достоверны по t-критерию Стьюдента на уровне 0,01.

Из таблицы видно, что девушки показали результат в среднем на 22% ниже юношей, студенты-горожане - на 12% выше сельских, а лицеисты – почти на 40% выше городских выпускников и на 30% выше выпускников сельских школ.

Данный феномен может быть объяснен следующим образом. Несмотря на превышение показателей анкеты у программистов, в целом, и у информатиков, в частности (см. табл. 2.9), уровень достоверности этих различий, вычисленный по t-критерию Стьюдента недостаточен для утверждения о достоверности. То есть по показателям анкеты конструктивной деятельности не обнаружено преимуществ одной категории студентов перед другими. Это означает уровень развития задатков конструктивной деятельности у всех студентов МФ был примерно одинаковым, но многие студенты, обучающиеся в группах ПМИ и ИС, еще в средних классах школы сориентировались на будущую специальность, что в сочетании с условиями, созданными в учебных заведениях и классах лицейского типа, а также с выраженной мотивацией, привело к развитию задатков до уровня выраженных способностей к программированию и формированию явно выраженной склонности к выбору программирования как профессии. В то же время у учащихся обычных городских и сельских школ, а также у большинства девушек, особенно у

математиков, указанные задатки развития не получили, что и отразилось на результатах спецтеста. Развить пусть даже сохранившиеся задатки в юношеском возрасте, в период студенчества, на наш взгляд, возможно только в специально созданных условиях.

9) На наш взгляд, следует осуществлять дифференциацию студентов на определенных рубежах обучения, т.е. вводить больше специализаций с различными учебными планами, позволяющими студентам своевременно выбрать тип профессиональной деятельности, специализацию в программировании.

10) В период подачи документов в приемную комиссию необходимо организовать полное и достоверное информирование абитуриентов и их родителей о содержании специальностей, об их профессиональной и социальной перспективе (через стендовую информацию, встречи с преподавателями и специалистами, групповые и индивидуальные консультации), с тем, чтобы подготовить их осмысленный выбор;

11) Для абитуриентов, успешно сдавших вступительные экзамены и допущенных к собеседованию для зачисления на факультет, следует организовать дополнительные консультации для уточнения их выбора, для формирования учебных групп и подгрупп. Последнее необходимо для выравнивания уровня подготовки по программированию и для обеспечения эффективности практических занятий. Принцип объединения в учебные группы и подгруппы должен быть не по изучаемому иностранному языку, как это делается сейчас, а по актуальному уровню компьютерной подготовки с учетом трех выявленных уровней - "лицейский", "городской", "сельский";

12) Организационно-методическое обеспечение занятий для групп и подгрупп указанных трех типов должно учитывать существенные различия в теоретической и практической подготовке их в области компьютерной грамотности и программировании;

13) Необходимо организовать выпуск вспомогательной учебно-методических и справочных изданий, рассчитанных на три уровня подготовленности первокурсников;

14) Для студентов всех специальностей и специализаций математического факультета во втором семестре необходимо проводить диагностическое обследование с целью составления прогноза успешности их подготовки по программированию с тем,

чтобы в дальнейшем готовить их по отдельным групповым учебным планам, через систему спецкурсов по выбору, как системных программистов, администраторов компьютерных сетей и баз данных, преподавателей "Основ информатики и ВТ" для учебных заведений разного уровня, специалистов по менеджменту и маркетингу программного обеспечения и других. Все эти категории специалистов будут востребованы на современном рынке труда.

Результаты анализа показали, что особенно чувствительными периодами для развития КЛМ являются:

а) ранний дошкольный возраст, когда придается начальный импульс развитию природных задатков (здесь, на наш взгляд, закладывается будущее «отставание» девочек);

б) средние классы школы, когда начинается системное изучение основ наук – в этот период учащиеся начинают формировать КЛМ на регулярных уроках математики, в то время как большинство из них еще либо не начали знакомство с ИВТ, либо делают это в преимущественно игровой форме. Таким образом, у ученика средних классов начинает формироваться математическое мышление, а мышление программиста не пробуждается. Позже – в старших классах – сформированная математическая компетентность станет существенным сдерживающим фактором в развитии КЛМ по программистскому типу.

Таким образом, цели констатирующего эксперимента - подготовка формирующего и контрольного этапов эксперимента достигнуты, а именно:

1) отработана процедура предварительной оценки исходного уровня развития конструктивно-логического мышления студента;

2) опробована измерительная процедура для оценки успешности развивающего обучения программированию и методики ее применения.

Констатирующий этап опытно-экспериментальной работы позволил выявить исходный уровень конструктивно-логического мышления студентов перед началом изучения основ программирования (ОП).

Результаты констатирующего эксперимента показали незначительное количество студентов, достигших третьего уровня сформированности конструктивно-логического мышления (2%),

четвертого и пятого уровней никто достигли единицы. Полученные результаты обусловили необходимость целенаправленной работы с использованием возможностей обучения программирования в развитии КЛМ-АСМ.

2.2. Методическое обеспечение и организация формирующего эксперимента. Анализ и интерпретация результатов формирующего эксперимента

Обучающий, формирующий эксперимент (2004-2010 гг.) состоял в обучении двух групп студентов: первой - по экспериментальной программе развивающего обучения программированию (экспериментальная группа - ЭГ), второй - по существующей программе (контрольная группа - КГ);

Здесь ставилась задача проверки эффективности развивающей программы обучения программированию, то есть проверки рабочей гипотезы о том, что системное применение базовых принципов обучения способствует существенному повышению эффективности обучения алгоритмизации и программированию, а значит развитию комплекса КЛМ-АСМ.

Типовая схема эксперимента: вся выборка студентов делилась на две примерно равные по начальному уровню группы – контрольную и экспериментальную (принцип деления!), и со контрольной группой проводилась дополнительная работа, направленная на интенсификацию развития КЛМ-АСМ, т.е. студенты этой группы были обязаны выполнять дополнительные задания. Студенты тщательно подбирались в пары, в которых один студент принадлежал к экспериментальной группе, а второй – к контрольной. Задачи для программирования и решения на ЭВМ они получали одинаковые, но объем выполняемой работы был разным. То, что «контрольный» студент делал в рамках современной «редуцированной» технологии обучения, это же «экспериментальный» студент выполнял в соответствии с целями и требованиями эксперимента, сдавая письменные и графические работы. Стоит отметить, что, в некоторых случаях, мотивация для участия в эксперименте в целом была выражена слабо, отдельные «экспериментальные» студенты считали себя обиженными и перегруженными, завидуя своим напарникам. Тем не менее, результат их дополнительных усилий был весьма существенным.

Перед началом работы, в соответствии с выводами констатирующего эксперимента, по каждому студенту с помощью анкет собиралась информация двух видов.

1. Сведения о стаже программирования и ранее достигнутой компетенции студента в области информатики и вычислительной техники в период обучения в школе; здесь собиралась информация такого рода:

- тип учебного заведения, который окончил студент (средняя школа – лицей, городская, школа райцентра, сельская; среднее специальное учебное заведение);
- возраст, когда школьник-будущий студент - впервые познакомился с компьютерной техникой;
- периоды изучения основ ИВТ, в целом, и основ алгоритмизации и программирования, в частности.
- уровень оснащенности учебного процесса по ИВТ в учебном заведении, где учился студент;
- какие педагогические средства, в частности, языки программирования использовались в конкретной школе. Например, на наш взгляд, язык Паскаль обладает существенно большим потенциалом для развития КЛМ, чем более простой и доступный язык Бэйсик. В то же время, наиболее популярный профессиональный язык программирования Си, как и его объектно-ориентированный вариант Си++, на наш взгляд, имеет специфику, не совместимую с принципами обучения в школьном возрасте. По каждому студенту составлялась запись следующего вида, например, «Б5-П9-11», что означало – будущий студент в период с 5-го по 8-й класс средней школы изучал язык Бэйсик, а в старших классах три года осваивал язык Паскаль.

2. Наличие у студента собственного персонального компьютера, степень доступности сетевых ресурсов. Такая информация позволяет обоснованно разделить студентов на примерно симметричные пары, в которых один студент представлял контрольную группу, а второй входил в экспериментальную группу. Оценивался уровень каждой пары (А – низкий, начинающие программировать; В – знающие основы и продолжающие изучение программирования; С – совершенствующиеся), в соответствии с которым студентам пары ЭГ-КГ выдавались задания-проекты для решения. Студенты, составляющие пару, получали строго одинаковые задания – задачи

для самостоятельного программирования, которые студент из контрольной группы выполнял по стандартной процедуре и отчитывался по стандартным требованиям, а студент из экспериментальной группы делал те же задания в соответствии с условиями эксперимента и отчитывался подробнее и в письменной графической форме.

Таким образом, при проведении практических занятий по программированию выборка обучающихся была поделена на две группы равной численности - контрольную (КГ) и экспериментальную (ЭГ). Студенты из разных групп были объединены в однородные пары по гендерному признаку, по стажу и качеству изучения информатики и программирования в школе. Будем обозначать студентов - участников одной пары, СКГ и СЭГ.

Для каждой пары СКГ-СЭГ на одном шаге эксперимента выдавалось одно задание-проект – задача для составления программ, на которую отводилось две-три недели. Выполнение задания участниками пары не синхронизировалось, тем более, что объем работы у участника ЭГ больше.

Студент КГ делал задание в режиме определяемом учебным планом, то есть он мог выполнять работу в учебной лаборатории, но для продолжительной работы с компьютером, ни времени, ни места, ему не отводилось. СКГ отчитывался о работе предъявлением текста программы на языке программирования и выполнением программы на тестовых данных.

Студент ЭГ не имел преимуществ при работе в компьютерном классе, он также, как и его напарник-студент контрольной группы, в основном работал самостоятельно, вне вуза, и, как правило, дома, на личном ПК. Объем работы СЭГ существенно больше – он должен был письменно-графически отчитаться в проведении всех этапов решения задачи, предъявить структурные схемы задачи, данных и действий, используемых при ее решении; особый акцент делался на этап алгоритмизации – составление алгоритма, реализующего метод решения задачи, подготовленный на предыдущих этапах работы. Таким образом текст программы у СЭГ становился естественным промежуточным звеном в технологической цепочке решения задачи, тем более, что СЭГ отчитывался по заданию в учебной лаборатории не простым предъявлением текста и прогоном программы на тестовых данных, но также пошаговым выполнением программы с заданием контрольных точек останова и контролируемых элементов задачи, а

также с устным объяснением демонстрируемого решения на экране ПК.

2.3. Контролирующий этап - проверка эффективности дидактических условий развития конструктивно-логического мышления

Контролирующий этап включал в себя:

- измерение уровня успешности в изучении основ программирования в обеих группах – дважды, после первого семестра обучения и после второго;

- обработку, анализ и интерпретацию результатов измерений (изменение уровня успешности в освоении основ алгоритмизации и программирования как показатель развития КЛМ-АСМ).

Промежуточное и итоговое тестирование с помощью процедуры измерения уровня успешности освоения студентами основ алгоритмизации и программирования.

При проведении процедуры студентам раздавались бланки с тестовыми заданиями (см. Приложение 3) и бланки для ответов и бумага для черновиков, так как ряд заданий требовали определенных черновых записей, анализа вариантов. Для корректности сравнений на процедуру всегда отводилось время одной академической пары часов, после чего выполняется ввод и обработка результатов каждого студента с помощью специально разработанной электронной таблицы в приложении Excel среды Windows. Оценивались:

- сумма баллов, набранных студентом;
- среднее значение для всей академической группы;
- средние значения для контрольной и экспериментальной групп;
- средние значения для различных уровней компетенции;
- средние значения для юношей и девушек.

Подчеркнем, что процедура тестирования к студентам обеих групп применялась двукратно – после первого и после второго семестра, то есть временной промежуток от начала эксперимента, и между двумя точками измерения составлял примерно полгода, что достаточно для оценки изменения состояния психологического новообразования. Результаты, усредненные по группам, приведены на диаграмме.

Из приведенных диаграмм видно, что незначительное превышение в уровне успешности выполнения теста экспериментальной группы над контрольной 7,9%, обозначившееся после первого семестра, в конце года становится очевидным и существенным – 21,5%.

Еще важнее распределение студентов двух групп по уровням компетенции:

- в контрольной группе до формирующего эксперимента 48% процентов студентов входили в группы начального и низкого уровня, но и по окончании эксперимента 30% остались в той же категории;

- в экспериментальной группе, перед началом эксперимента те же 47,8% студентов были в нижних группах компетенции, но по результатам формирующего эксперимента там осталось только 13%;

Из приведенных диаграмм видно, что незначительное превышение в уровне успешности выполнения теста экспериментальной группы над контрольной 7,9%, обозначившееся после первого семестра, в конце года становится очевидным и существенным – 21,5%.

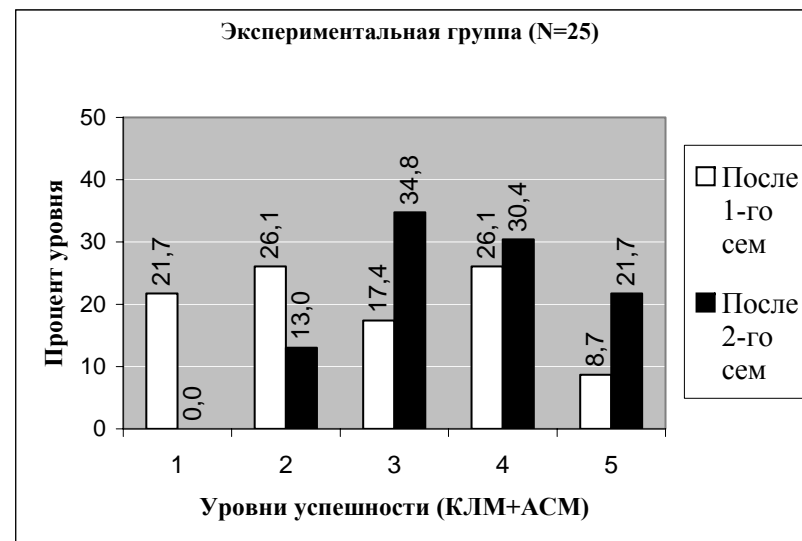
Еще важнее распределение студентов двух групп по уровням компетенции:

- в контрольной группе до формирующего эксперимента 48% процентов студентов входили в группы начального и низкого уровня, но и по окончании эксперимента 30% остались в той же категории;

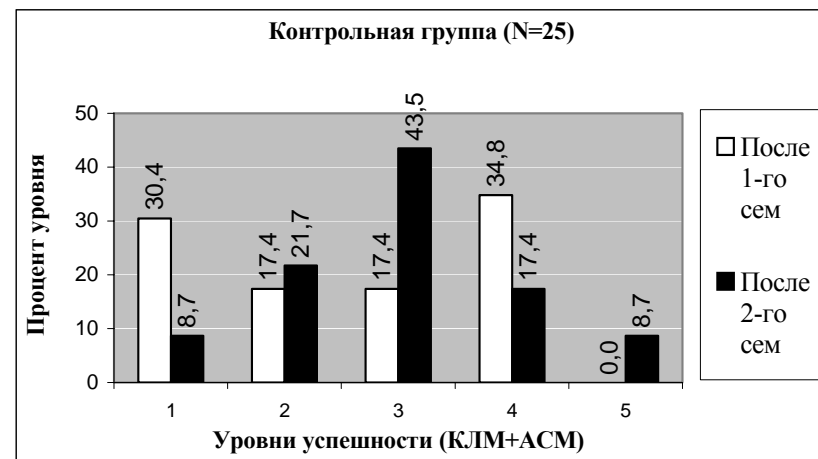
- в экспериментальной группе, перед началом эксперимента те же 47,8% студентов были в нижних группах компетенции, но по результатам формирующего эксперимента там осталось только 13%;

- половина успешных студенты контрольной группы на начало второго семестра (34,8%) в конце перераспределились между двумя категориями – ниже и выше по компетенции, дав в сумме 70%, в то же время в экспериментальной группе эти же три категории дали в сумме 87%;

Динамика уровня развития КЛМ экспериментальной группы, %



Динамика уровня развития КЛМ студентов контрольной группы, %



Интересно оценить – чего достигает студент той или иной категории в обеих группах в ходе обучения и почему?

Для двух групп студентов, участвовавших в эксперименте, выявлены пять схем развития конструктивно-логического мышления:

а) *студент КГ:*

а1) находился на низком начальном уровне развития КЛМ и, в частности, имел слабую школьную подготовку по информатике и ВТ – в этом случае преимущественно самостоятельная работа по общей программе с минимальным контролем, в лучшем случае, продвинул его на один шаг в сторону работы по образцу – с уровня 1 на уровень 2 (21,7%), а, в худшем, он останется в той же точке развития своего мышления, т.е. на уровне 1 или 2 (8,7%);

а2) обладал хорошим уровнем развития КЛМ, определяемым хорошей (как правило, лицейской) подготовкой по математике и информатике и ВТ – в этом случае обучение в рамках КГ переведет студента со 2-го уровня на 3-й уровень, но может снизить его показатели с 4-го уровня на 3-й (в сумме 43%);

а3) отдельные студенты КГ (8,7%) изначально показывали наиболее высокий 5-й уровень развития КЛМ-АСМ, но в конце экспериментального срока они снизили свои показатели до 4-го уровня;

б) *студент ЭГ:*

б1) обладал изначально низким уровнем развития КЛМ и, в частности, имел слабую школьную подготовку по информатике и ВТ – в этом случае работа по экспериментальной программе перевела 47% изначально слабых студентов (уровни 1 и 2) в разряд хорошо развитых в смысле КЛМ-АСМ (48%);

б2) 52% студентов ЭГ, показавших после 1-го семестра средний (3-й), хороший (4-й) и высокий (5-й) уровни развития комплекса КЛМ-АСМ (это, как правило, выпускники специализированных математических классов обычных школ, не имеющие выраженной компьютерной подготовки), в конце экспериментального обучения образовали 52% студентов с высоким и хорошим уровнями КЛМ-АСМ; обращает на себя внимание почти 3-х-кратное увеличение - с 8,7% до 21,7% - доли студентов, достигших высокой успешности в алгоритмизации и программировании и продвинутого уровня развития комплекса КЛМ-АСМ.

Можно сделать вывод, что экспериментальная методика наиболее эффективна именно для студентов с изначально низким уровнем развития КЛМ-АСМ, но именно эти категории обучаемых, которые заведомо не нацелены на карьеру программиста и, при этом, составляют большую часть студентов непрограммистских естественнонаучных специальностей, и являются объектом нашего исследования и развития.

Экспериментальная проверка разработанной теоретической модели процесса развития конструктивно-логического мышления студентов показала, что сформированные в результате экспериментального обучения конструктивно-логические операции удовлетворяют всем требованиям, предъявляемым теорией к умственным действиям и знаниям:

- **об обобщенности** умений по применению операций свидетельствует тот факт, что изучение алгоритмизации и программирования способствовало самостоятельному решению новых задач;

- **о развернутости** операций говорит тот факт, что учащиеся в ходе программирования задач представляли все более подробные и обстоятельные письменные протоколы с описанием структуры задач и структуры данных, граф-схем алгоритмов, как отдельных подзадач, так и задачи в целом;

- **о мере освоения** операций говорит быстрота их выполнения, которая заметно увеличилась у студентов экспериментальной группы, что дало возможность решать за урок большее число задач;

- **о разумности** выполняемых операций можно судить по специально подобранным задачам, требующим самостоятельного анализа и синтеза информации и использования всех алгоритмов для поиска решения;

- сформированные конструктивно-логические операции являлись **осознанными**, так как студенты продемонстрировали понимание назначения каждой операции, то есть ту конечную цель, которую им предстоит осуществить.

В ходе экспериментального исследования было установлено, что успешное развитие конструктивно-логического мышления студентов находится в прямой зависимости от оптимального сочетания коллективной и индивидуальной деятельности обучаемых.

Сопоставление результатов констатирующего и формирующего экспериментов (диаграммы 2.1 и 2.2 на стр. 74) показало, что реализация созданной экспериментальной методики развивающего обучения позволила существенно повысить уровень развития конструктивно-логического мышления студентов на занятиях по алгоритмизации и программированию. Результаты экспериментального исследования показали, что большинство студентов экспериментальной группы, достигли среднего и высокого уровней развития конструктивно-логического мышления.

В ходе исследования было выявлено, что активизация мыслительной деятельности студентов экспериментальной группы осуществляется главным образом с помощью дополнительных заданий на анализ структуры задачи, данных и алгоритмов, в процессе выполнения которых у студентов, вначале выполняются принудительно, а затем, по мере освоения деятельности и выработки навыков, осознанно вырабатывается не только алгоритмический стиль мышления, но и, у лучших студентов, элементы профессионального стиля программирования.

Изучение учебных курсов «Компьютерные науки», «Искусство программирования», «Алгоритмизация и языки программирования» и т.п. студентами объективно необходимо для успешного развития конструктивно-логического мышления студентов, так как оно способствует развитию их познавательной активности и самостоятельности, формированию профессиональных качеств. Все студенты, принявшие участие в исследовании, овладели профессиональной методикой программирования и решения задач с помощью персонального компьютера, элементами промышленной технологии создания программного обеспечения.

Обучение программированию способствует воспитанию у студентов внутренней потребности в знаниях, желанию учиться, стремлению к познанию, получению удовлетворения от преодоления трудностей. Знания приобретают для студентов большую значимость, так как они могут их использовать в практике решения задач с использованием современных программных инструментов. У студентов возникает познавательный интерес и помимо развития конструктивно-логического мышления у них развивается творческое мышление.

2.4. Рекомендации по обучению будущих педагогов-преподавателей математики и информатики

Проведенные теоретические и экспериментальные исследования позволяют сформулировать рекомендации для обучения основам программирования студентов естественнонаучных специальностей на младших курсах вуза.

Данные рекомендации вытекают из личного педагогического опыта автора, а также из опыта и концептуальных работ ведущих профессоров прикладной математики и информатики математического факультета Удмуртского госуниверситета А.П.Бельтюкова [4] и Н.Н.Непейводы [91], а также из обсуждений опыта и перспектив математического и информационного образования, как на семинарах по теоретической информатике математического факультета УдГУ, так и на рабочем совещании «Преподавание информатики как комплексная проблема» [103]. Ключевой момент всех обсуждений и принятых рекомендательных решений заключается в пересмотре стандартов и всей системы подготовки специалистов по прикладной математике, информатике и программированию, а значит и коренной перестройке преподавания этих наук и дисциплин.

В настоящее время обучение программированию на младших курсах математического факультета Удмуртского госуниверситета строится на принципах усвоения базового комплекса знаний, выработки умений и навыков программиста, на основе которых в последующем ведется профессиональная подготовка, дифференцированная для отдельных специальностей. При этом в соответствии с выбранной специальностью студенты получают три уровня подготовки по программированию:

1-й уровень – специалист предметной области-квалифицированный пользователь, начинающий программист, а также преподаватель основ информатики и ВТ (специальности – «математик, преподаватель математики»; сюда можно отнести и специализацию «компьютерный дизайн» специальности «информатик»; бакалавр математики и прикладной математики);

2-й уровень – прикладной математик и системный программист (специальность – «прикладная математика и информатика»);

3-й уровень – информатик-системотехник (специальность – «информационные системы»).

На первом курсе начала программирования преподаются в рамках следующих дисциплин: "ЭВМ, информатика и практикум по программированию" (ныне «Компьютерные науки» для математиков-педагогов и бакалавров), "Практикум на ЭВМ" и "Языки программирования и методы трансляции" (для прикладных математиков), "Введение в информатику" и "ЭВМ и искусство программирования" (для информатиков), но учебные часы отведены примерно в равном объеме - 72 часа в 1-м семестре и столько же во втором. При этом наполнение часов различно.

Математики изучают основы компьютерной грамотности, алгоритмизации и программирования задачи на языке Паскаль. Раз в две недели на семинаре они практикуются в детальном анализе учебных задач, синтезе алгоритмов и их кодировании на Паскале. Получая реально два часа лабораторного практикума в неделю, они занимаются в компьютерном классе Интернет-центра университета, где приобретают необходимые умения и навыки работы с ПЭВМ, операционной графической оболочкой Windows NT системой программирования Турбо Паскаль. При этом выдерживается следующий темп учебных занятий – студент за одну-две недели должен самостоятельно выполнить одно задание - законченную программу, и продемонстрировать ее в работе. Сложность заданий нарастает одновременно со сложностью тем лекционного материала, и соответствует нарастающему уровню комплекса КЛМ-АСМ.

Прикладные математики, как и информатики, большинство из которых уже со школьных лет сориентировано на компьютерные специальности и имеет хорошую базовую подготовку, занимаются по усложненной программе. Они должны владеть, как минимум, одним языком программирования (Бейсик или Паскаль), а лучше - обоими. Фактически, их не обучают началам программирования, и с первых дней они совершенствуют имеющиеся ЗУН. Начальные задания для их самостоятельной работы примерно соответствуют уровню сложности заданий, выполняемых математиками в конце 1-го курса. Прикладные программисты изучают на 1-м курсе второй (или третий) язык программирования (язык Си), а информатики - два языка (Си и Ассемблер). То есть темп обучения программированию у всех трех специальностей различен.

На наш взгляд, первокурснику, пришедшему в вуз из сельской или городской периферийной школы, где практически отсутствует преподавание основ информатики и ВТ на должном уровне,

освоить программу специальностей ПМИ и ИС за первый курс весьма сложно. У таких студентов нередко отсутствуют даже начальные знания и умения работы с компьютером, не говоря уже о развитых умениях и закрепленных навыках. Отсюда выполнение ими учебных заданий растягивается на чрезмерный срок, студенты выпадают из учебного графика, получают отрицательные оценки в аттестацию и на зачете, что неизбежно сказывается на общей успешности их адаптации и обучения в вузе.

Контроль за успешностью обучения программированию в настоящее время строится по принятому в университете стандарту - две аттестации в ходе семестра и зачет в конце семестра. Требуемый индивидуальный объем работы (сложность заданий) для каждого студента варьируется в определенных пределах и устанавливается преподавателем в зависимости от актуального уровня подготовки каждого студента с учетом требуемого минимума подготовки. Очевидно, что потенциал каждого студента (его зоны актуального и ближайшего развития) различен, и его необходимо принимать во внимание.

Экзамен по программированию на первом курсе проводится один раз в конце учебного года, а итоговая отчетность имеет форму зачета-отчета по заданиям, выполненным в течение семестра. То есть система объективного количественного контроля уровня ЗУН по программированию на 1-ом курсе отсутствует, и настоящее исследование является первой попыткой разработки инструмента для объективной оценки.

Создание батареи специальных тестов по информатике и программированию могло бы восполнить этот пробел и дать необходимые индикаторы для преподавателя и ориентиры для студента. Такое тестирование можно было бы проводить раз в две недели или приурочивать к очередной аттестации.

В качестве примера рассмотрим математический факультет Удмуртского госуниверситета как педагогическую систему.

В основе готовности к решению профессиональных задач (какие профессии (5 – М, М.ПМ, ПМИ, ПИ, ММЭ) лежит, в частности, уровень развития КЛМ-АСМ. У педагогов задача сложнее – в ПС средних учебных заведений уметь сформировать основы КЛМ-АСМ у будущих специалистов, бакалавров или магистров.

Для удобства введем следующие обозначения: педагогическая система обучения математике и информатике или педагогическая система математического факультета УдГУ (МФ УдГУ) – ПС МФ.

Анализ учебных планов показал, что ПС МФ может быть поделена на три части, соответствующие трем основным подсистемам подготовки студентов – по математике, по информатике и программированию, педагогическая подготовка.

ПС МФ = ПС обучения математике + ПС обучения основам алгоритмизации и программирования студентов университета (на примере МФ, с точки зрения развития КЛМ и АСМ) + ПС основ психологии и педагогики.

«Старый» и «новый» стандарты специальностей. Часы в учебных планах. Налицо парадокс – чем больше компьютеров, тем хуже учим программированию. Наступает перекося – студент стремится к компьютеру (операционная направленность), он торопится написать программу, оставляя позади самое главное – анализ задачи, поиск идеи, построение алгоритма. Здесь нарушается принцип научности, системности.

Все больше прикладных пакетов программ и доступных систем программирования, отсюда все больше времени отводится и тратится на их изучение – обучение идет вширь, но не вглубь. Здесь нарушаются те же принципы.

Анализ документов и действующей практики ПС МФ показывает, что эти три подсистемы существуют во многом раздельно и независимо, они весьма слабо связаны между собой, нет их явного взаимодействия. То есть, вместо того, чтобы развиваться в качестве взаимосвязанных, взаимодействующих элементов – частей единого целого, эти три составляющие существуют как почти независимые подсистемы, лишь механически объединенные в организационных рамках ПС МФ. Это не может не сказываться на результативности процесса обучения, в целом, и в частности, в интересующем нас аспекте – развитии КЛМ-АСМ.

Варианты организации обучения по математике и программированию приведены на следующем рисунке:

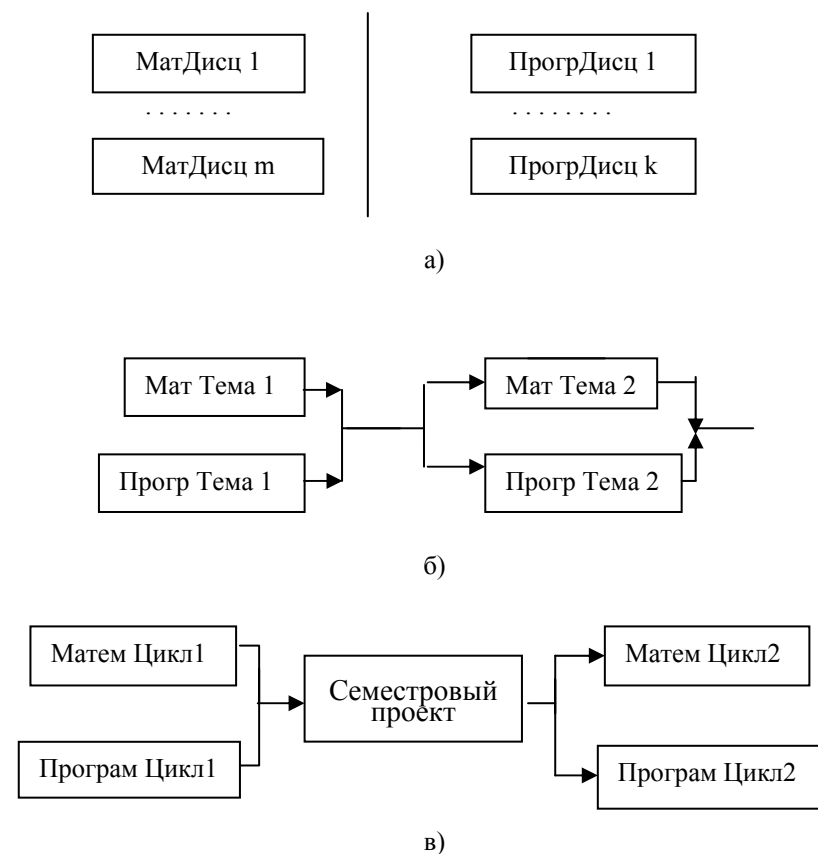


Рис.2.1. Варианты организации совместного обучения математике и программированию: а) действующий; б) с синхронизацией по темам; в) с синхронизацией по циклам.

В последний вариант для математиков-педагогов в семестровый курсовой проект целесообразно включать психолого-педагогические разделы, например, методическую проработку темы или цикла.

Как уже отмечалось ранее, параллельно с обучением алгоритмизации и программированию данная категория студентов

проходит обучение математике, то есть имеет место сложное взаимодействие двух существенных факторов развития КЛМ.

Может быть сформулирована следующая гипотеза: при асинхронном неравномерном развитии двух взаимодействующих видов компетенции, особенно на ранних стадиях, они препятствуют развитию друг друга. При доминировании одной из них, вторая подавляется, задерживается в развитии, в результате, хорошо подготовленный математик уже не станет продвинутым программистом (но может стать отличным аналитиком, постановщиком задач), а высококвалифицированный программист уже никогда не станет математиком. В то же время, при синхронном продвижении по двум параллельным направлениям, математическая и программистская компетенции способствуют развитию друг друга, и при определенном уровне математической и программистской компетенции эти два фактора не просто складываются, а обогащают, стимулируют друг друга.

Сказанное выше, по нашему мнению, должно быть отражено в методике преподавания математикам двух основных циклов дисциплин, относящихся к математике и информатике.

Рассматривая математический факультет Удмуртского университета как педагогическую систему, отметим, что в основе готовности к решению профессиональных задач математики и информатики лежит, в частности, комплекс КЛМ-АСМ. Именно на развитие этого комплекса и должно быть направлено обучение. Для категории студентов - будущих педагогов задача сложнее: в рамках педагогических систем различных уровней общего и профессионального образования они должны в будущем организовать обучение и формирование названных качеств у будущих работников или студентов профессионального учебного заведения.

В соответствии с подходом Н.В. Кузьминой педагогическую систему математического факультета (ПС МФ) можно представить как объединение трех подсистем (ПдС): ПдС обучения математике; ПдС обучения информатике и программированию; ПдС психолого-педагогической подготовки.

Анализ руководящих и рабочих документов, действующей практики ПС МФ показывает, что эти три подсистемы существуют во многом раздельно и независимо, они весьма слабо связаны между собой, между ними нет явного взаимодействия. То есть, вместо того, чтобы работать и развиваться в качестве

взаимосвязанных, взаимодействующих элементов – частей единого целого, эти три составляющие существуют как практически независимые подсистемы, лишь механически объединенные в организационных рамках ПС МФ. Это не может не сказываться на результативности процесса обучения, в целом, и в частности, в интересующем нас аспекте – развитии КЛМ-АСМ.

В этой связи отдельного внимания заслуживает вопрос совместного обучения математике и программированию, их синхронизации. В нашем исследовании нашла подтверждение гипотеза о том, что две компетенции у обучаемого – математическая и программистская, формируясь одновременно, находятся в сложном динамическом взаимодействии. При асинхронном неравномерном развитии взаимодействующих видов компетенции, особенно на ранних стадиях, они препятствуют развитию друг друга. При доминировании одной из них, вторая подавляется, задерживается в развитии, в результате, хорошо подготовленный математик с большой вероятностью уже не станет продвинутым программистом, и, в свою очередь, квалифицированный программист со стажем уже никогда не проявит интереса к математике.

На наш взгляд, наилучшим вариантом видится такой, когда развитие двух компетенций согласованно, синхронно, и ни одна из них не берет верх, подавляя другую. В противном случае, по-видимому, происходит своеобразный захват одной компетенцией единого конструктивного и логического ресурса мышления у обучаемого, так что развитие второй компетенции заметно тормозится. В то же время, при синхронном продвижении по двум параллельным направлениям, при условии правильного методического подхода, математическая и программистская компетенции опираются друг на друга, способствуют развитию друг друга, и на определенном уровне эти два фактора не просто складываются, но и умножаются, обогащают друг друга. В этом случае можно говорить о достижении важной цели – воспитании высококвалифицированного прикладного математика, информатика и программиста в одном лице, который способен находить, ставить и формализовать решение задач в любой предметной области, доводить решение этой задачи до высококачественного программного продукта.

Вопрос использования конструктивно-логического ресурса в развитии двух конкурирующих компетенций требует более

глубокой проработки, с тем, чтобы не просто констатировать факт жесткого распределения ресурса, но и обозначить пути возможной корректировки сложившегося состояния, с тем, чтобы исключить крайние ситуации, когда вчерашний школьник с большим компьютерным стажем, придя в вуз, не может освоить программу по математике, а студент, успешный в математике, оказывается беспомощным у компьютера - программный конструктивизм ему недоступен. И здесь сможет найти свое воплощение дидактический принцип индивидуальности обучения.

Сказанное может быть отражено в организации и методике преподавания математикам двух основных циклов дисциплин, относящихся к математике и информатике. Именно развитие комплекса КЛМ-АСМ могло бы стать стержнем, на котором может быть выстроен процесс подготовки всех математических специальностей, как и других специальностей, безусловно опирающихся на конструктивность и логику.

Возможны следующие варианты организации обучения по математике и программированию: а) действующий – математические дисциплины и предметы информационного цикла изучаются параллельно и независимо; б) синхронизация по темам; в) синхронизация по циклам. В последний вариант для математиков-педагогов целесообразно в семестровый проект включать частнодидактические, психоло-педагогические или методические разделы, например, методическую проработку темы или цикла, в которых явно соединяются математика и информатика.

Еще один путь перестройки учебного процесса в соответствии с современными требованиями к обучению математиков и программистов состоит, на наш взгляд, в разработке специального учебного языка программирования и его системы программирования – методической основы, объединяющей конструктивные линии математики и программирования, но это направление еще потребует отдельной дидактической и методической проработки.

Предлагая варианты перестройки процесса обучения математике и информатике в рамках единой ПС, согласно системной модели педагогической деятельности Н.В.Кузьминой [76], следует сказать и о субъектах этой деятельности – если студенты, поступающие на специальности математического факультета и родственные им по характеру обучения и деятельности, достаточно однородны по своим качествам, то педагогический состав четко делится на три

группы, соответствующие вышеназванным педагогическим подсистемам. При этом психолого-педагогическая ПдС, в своей основной части, выходит за пределы ПС факультета, а значит и весьма слабо взаимодействует с двумя другими, которые также слабо взаимодействуют. Таким образом, и здесь резервы совершенствования процесса обучения тоже велики.

Особый значение проведенные исследования и их результаты, на наш взгляд, имеют для обучения студентов специальности «Математик. Преподаватель математики», чья текущая и будущая учебная деятельность по отношению к себе и по отношению к учащимся, соответственно, нацелена на саморазвитие и развитие личности и на формирование мышления учащихся, его отдельных сторон.

Выводы

В ходе трех этапов экспериментальной работы развитию конструктивно-логического мышления у студентов-математиков решались различные комплексы задач. На первом этапе – в констатирующем эксперименте – отработана методика и измерительные процедуры по оценке состояния и развития комплекса КЛМ-АСМ у студентов-математиков, изучающих математику и основы программирования на младших курсах вуза. На втором этапе – в формирующем эксперименте – была использована разработанная на основе модели обучения экспериментальная программа обучения основам алгоритмизации и программирования, что дало существенный эффект в успешности освоения студентами основ программирования, а значит и в развитии их конструктивно-логического мышления и освоения ими алгоритмического стиля мышления.

Анализ хода и результатов экспериментов показали, что экспериментальная методика, опирающаяся на фундаментализацию обучения программированию и системную реализацию как общедидактических принципов обучения, так и частнодидактических подходов к обучению программированию, стимулирует активную позицию студента, формирует в нем интерес к освоению программирования, что отражается на результатах обучения.

В ходе экспериментальной работы:

а) на констатирующем этапе: отслеживались показатели общего интеллектуального развития студентов-математиков и показатель их успешности в изучении основ алгоритмизации и программирования; зафиксировано, что:

- показатели развития общих познавательных способностей практически одинаковы для всех категорий студентов-математиков. Это подтверждает, что на естественнонаучные специальности поступают выпускники школ, сориентированные на них еще во время обучения в школе, и фактор случайности очень незначителен;

- по результатам специального теста, которые показали студенты-математики после изучения основ программирования, все студенты делятся на три неравные группы – высокой успешности, средней успешности и низкой успешности в обучении программированию; из таблиц видно, что у математиков по четверти студентов показали высокий и низкий результаты, а ровно половина студентов – девушки и юноши, показали среднюю успешность, тогда как у будущих программистов больше половины девушек показали высокий результат и по четверти – средний и низкий, и 88% юношей – высокий, и только 1 студент – низкий; т.е. показатели программистов в 2,5 раза превосходят аналогичные показатели математиков, показатели юношей почти в 2 раза выше, чем у девушек;

б) на формирующем и контрольном этапах отслеживался один показатель - успешность освоения основ алгоритмизации и программирования; зафиксировано, что успешность обучения программированию студентов, оцениваемая с помощью специальной измерительной процедуры, у студентов экспериментальной группы (ЭГ) существенно (на 15-20%) выше, чем у студентов контрольной группы (КГ).

Анализ результатов опытно-экспериментальной работы свидетельствует об эффективности выявленных дидактических условий развития конструктивно-логического мышления и формирования алгоритмического стиля мышления в процессе изучения программирования.

Проведенные теоретические и экспериментальные исследования позволили сформулировать организационно-методические рекомендации по обучению основам программирования студентов естественнонаучных специальностей на младших курсах вуза.

Анализ научно-педагогической литературы [47-48], посвященной практическому мышлению и его развитию, позволил выдвинуть гипотезу, что существенное влияние на развитие такой разновидности практического мышления как конструктивно-логическое мышление у студентов может быть достигнуто посредством обучения их программированию задач для ЭВМ.

Исследование показало, что дидактически обоснованная модель процесса развития конструктивно-логического мышления у студентов строится в рамках алгоритмической парадигмы программирования путем согласования логики освоения основ программирования с логикой развития конструктивно-логического мышления, с учетом исходного уровня развития этого вида мышления.

Установлено, что согласование логики развития конструктивно-логического мышления с логикой обучения программированию, системное соблюдение ключевых дидактических принципов и их реализация в предложенной модели обучения, в конкретных методах и приемах, существенно повышают успешность обучения программированию, а следовательно существенно развивают конструктивно-логическое мышление студента.

Успешность процесса развития конструктивно-логического мышления у студентов, изучающих основы алгоритмизации и программирования, достигается благодаря реализации преподавателем комплекса дидактических условий, включающего: использование задания-проекта как формы структурирования учебного материала; логическую законченность каждого задания-проекта, связывающего определенный этап освоения логики программирования с этапом развития конструктивно-логического мышления у студента; использование в учебной деятельности активных методов обучения; использование в учебной деятельности активных форм организации учебного процесса, в частности, специально организованной самостоятельной работы студента как основной формы; партнерские отношения субъектов учебной деятельности, как студента и преподавателя, так в парах студентов, выполняющих одинаковые задания.

Реализация авторской программы обучения студентов основам алгоритмизации и программирования позволила зафиксировать положительную динамику успешности в изучении программирования как показателя развития конструктивно-логического мышления. Тем самым подтвердилась гипотеза о том,

что существенное развитие конструктивно-логического мышления у студентов в процессе обучения программированию обусловлено внедрением в учебный процесс специально разработанного комплекса дидактических условий.

Проведенное исследование выявило ряд новых проблем, требующих решения. В рамках дальнейшего изучения конструктивно-логического мышления студентов как разновидности практического мышления необходимо:

а) продолжить теоретическое и экспериментальное исследование конструктивно-логического потенциала различных видов учебной деятельности обучающихся и практической деятельности специалистов современных отраслей экономики, а также возможностей по развитию КЛМ в рамках учебных планов вузовских специальностей и рабочих программ отдельных дисциплин.

б) в рамках известного идейного и генетического единства математики и программирования необходимо исследовать механизмы взаимодействия при обучении студента двух видов конструктивно-логической деятельности и соответственно двух компетенций – математической и программиста, что актуально для педагогов при обучении учащихся средней школы и студентов математических и компьютерных специальностей вузов;

в) в настоящей работе алгоритмизация задачи и программирование ее для решения на ЭВМ рассматривались не только как средство развития комплекса конструктивно-логического мышления, но и как конструктивный метод моделирования явлений действительности, т.е. как метод познания через построение программных моделей. В то же время, и само программирование как метод моделирования тоже не стоит на месте, а постоянно эволюционирует; перспективным направлением является исследование психолого-педагогических аспектов обучения студентов программному моделированию в рамках современных парадигм – объектно-ориентированной и парадигмы управления так называемыми бизнес-процессами предметной области.

Целесообразно продолжить исследование возможностей и специфики разработанного подхода при развивающем обучении информатике и основам управления, как в других возрастных периодах, так и на других уровнях и в других формах образования.

Глава 3. Элементы дидактической системы формирования конструктивно-логической компетентности: от выявления одаренности до начальных уровней профессионализма в информатике

3.1. Мышление информатика-программиста и управленческое мышление как виды практического мышления. Информационное моделирование и развитие управленческого мышления

Существенные признаки практическое мышление (ПМ) приведены в работах С.Л. Рубинштейна [110] и Б.М. Теплова [118]:

- а) связь с действием;
- б) связь с непосредственным практическим преобразованием мира;
- в) конкретность;
- г) временные ограничения в протекании мыслительных операций;
- д) существенная роль образно-чувствительных компонентов.

Отечественная психология выделяет три вида практического мышления – предметно-действенное (мышление рабочего), оперативное (мышление оператора технических и информационных систем) и управленческое (УМ). Последнее является предметом наших исследований.

Сегодня основным содержанием профессиональной деятельности работников в любой сфере деятельности является управление некоторой предметной областью через анализ и моделирование ее процессов и явлений конструктивно-логическими средствами (КЛС), для чего управленцу в любой профессиональной области необходим определенный уровень как управленческого мышления, так и конструктивно-логического мышления (КЛМ) – еще одной разновидности практического мышления, впервые описанного автором при анализе мыслительной деятельности программистов и обучающихся программированию.

Именно конструктивно-логическое мышление требуется учителю и ученому, инженеру и экономисту, рядовому менеджеру и руководителю организации при анализе, описании и оптимизации управляемой ситуации в предметной области, при принятии решений.

Конструктивно-логическое мышление требуется для эффективной работы с современными программными системами, поддерживающими деятельность широкого круга специалистов. Одним из наиболее эффективных средств разрешения управляемой

ситуации является программное моделирование, или, говоря по-другому, программирование задач для ЭВМ.

Глубинный смысл известной метафоры академика А.П. Ершова «Программирование – вторая грамотность» [23], сегодня может означать еще и то, что нельзя рассматривать всех специалистов, занятых программным моделированием и управлением в различных областях практики, как программистов – для них создаются специальные программные системы описания и анализа предметной области. Содержанием работы таких специалистов является сбор данных, анализ, описание и измерение параметров управляемой ситуации, ее структурирование на составляющие, конструирование различных вариантов управления и их оценка, выбор наиболее оптимального варианта решения.

Выделить общие моменты в деятельности управленца и программиста, и использовать развивающее обучение программному моделированию не как обучение студентов основам управления, а как средство формирования КЛМ и через него развитие УМ. В основу такого обучения, на наш взгляд, целесообразно положить синтез как известных концепций в психологии компьютеризации, разрабатываемых в рамках смысловой теории мышления О.К. Тихомирова [120-123] его учениками на факультете психологии Московского университета, а также подходов, предложенных Д.Н.Завалишиной [27-29] и А.М. Матюшкиным [80] для развития оперативного мышления и организации проблемного обучения, соответственно, а так же специфических психолого-педагогических средств, методик, разрабатываемых на факультете ФИТиВТ Удмуртского государственного университета.

Таблица 3.1
Мышление специалиста управления и информатика/программиста

Виды практического мышления	
УМ	МП – МИП
Схема УМ: «подготовка решения – его принятие - исполнение»	Схема МП при программировании задачи – на каждом шаге подсхему УМ – анализ структуры задачи - разработка алгоритма, схемы решения – программирование, создание исполняемого модуля задачи

Субъекты – решения и исполнения - не совпадают	Студент, программирующий задачу, един в трех лицах: постановщик, лицо, принимающее решение, исполнитель (заказчик - преподаватель)
Типы практических мыслительных задач:	
Задачи на проектирование и рационализацию технологии	Задачи на разработку алгоритмов и перестройку алгоритмов
Прогностические задачи, важнейшей формой которых являются задачи предварительного планирования деятельности, а результатом решения – план предстоящей деятельности как совокупность способов, средств и сроков выполнения определенных действий, приводящих к достижению цели	Анализ структуры задачи, планирование структур данных и разработка структуры алгоритма и составляющих подалгоритмов, планирование отладки и сборки модуля задачи
Диагностические задачи, состоящие в основном в установлении причин возникновения неполадок и брака, особенно в тех случаях, когда неполадка может вызываться несколькими причинами	Диагностика и поиск ошибок в алгоритме и программе – синтаксических, семантических, логических – отладка программы
Иногда к самостоятельным функциям ПМ относят контроль (самоконтроль). Но он включает лишь проверку и оценку собственных действий и их результата, т.е. не имеет на выходе самостоятельного продукта, отличного от продукта решения вышеперечисленных задач, а потому в контексте решения	В рамках занятий и самостоятельной работы самоконтроль студента дополняется контролем со стороны преподавателя

мыслительных задач, по содержанию всегда вплетен в реализацию и верификацию решения этих задач	
Объект регулирования и управления	
Управление коллективом или совокупностью подразделений	Управление коллективом разработчиков или совокупностью групп разработчиков

3.2. Проектирование содержания при обучении программированию. Микроанализ программирования как последовательности проблемных ситуаций управления

Будем рассматривать область организационного, социального или педагогического управления: целесообразно рассмотреть типовую типовые действия из практики работы специалиста-управленца – это действия по анализу, синтезу и принятию решения, по его реализации. В ходе управления деятельность исполнителя включает следующие этапы:

1) выявление и анализ проблемной ситуации (С) в предметной области (ПрО);

2) поиск идеи и способа разрешения ситуации;

3) выбор, структурирование и организация ресурсов (Р), имеющихся в его распоряжении и действий (Д) над этими ресурсами, переводящими ситуацию управления С последовательно из исходного состояния C_0 через промежуточные состояния C_1, C_2, \dots в требуемое конечное состояние C_k :

$$C: C_0 \longrightarrow C_1 \longrightarrow C_2 \longrightarrow \dots \longrightarrow C_k,$$

где через C_i , обозначена i -я промежуточная ситуация, являющаяся результатом поиска/построения и применения некоторой функции F к предшествующему состоянию ситуации C_{i-1} :

$$C_i = F(C_{i-1}, P_{i-1}),$$

где аргументом P , функции является совокупность ресурсов — вектор $P_{i-1} = \langle r_{i-1}^1, r_{i-1}^2, \dots, r_{i-1}^n \rangle$ на текущем этапе решения задачи управления ПрО.

Будем называть действием применение функции F по переводу ситуации из состояния C_i в некоторое состояние C_j :

$$D_{ij} : C_i \rightarrow C_j \quad ;$$

4) на практике ситуация управления может развиваться не только как показано в простой последовательности, но может также претерпевать разветвления, возвраты назад, повторяться циклически, допускать вложения.

5) в процессе решения задачи управления, во всех компонентах решения — ситуациях, ресурсах и действиях — исполнителю требуется компетенции (ЗУН) системного анализа и синтеза, позволяющие выявлять и использовать следующие характеристики структур и процессов, такие как регулярность, декомпозицию, вложенность, агрегаты, прототипы и др.

Значительный резерв повышения эффективности обучения заключается в более глубокой и детальной психолого-педагогической проработке процессов обучения и усвоения знаний при изучении программирования, например, в рамках концепции проблемного обучения, когда решение задачи организовано как поиск неизвестного, как обнаружение нового знания. С одной стороны, программирование и обучение программированию по сути являются проблемными задачами, но для эффективного обучения программированию и через него основам управления требуется организовать обучение алгоритмизации и программированию как проблемное обучение, то есть как решение системы проблемных задач в соответствии определенными теоретическими рекомендациями, выработанными, например, академиком РАО А.М. Матюшкиным в [80].

Нами использован подход А.М. Матюшкина для описания, создания и использования проблемных ситуаций в обучении, когда задача порождает для обучаемого проблемную ситуацию, решая которую, он активизирует свое мышление и извлекает новое знание.

Ниже приводится пример проектирования практических занятий по программированию в виде дидактической цепи заданий - системы проблемных задач. Изучаемая тема - «Регулярность. Регулярные действия и регулярные данные». Основная проблемная ситуация (ОПС): организация регулярных повторяющихся действий над регулярными однородными

данными. Промежуточные, вспомогательные проблемные ситуации (ВПС) указаны в пунктах таблицы.

Таблица 3.2

Пример дидактической цепи заданий-проектов по программированию как последовательности логически взаимосвязанных проблемных ситуаций

Задание-проект (ОПС) как последовательность проблемных ситуаций (ВПС) (схема решения)	Программные конструкции и ресурсы	Действия (Δ – новое знание)
0. Задать константу /ввести число и вывести его на печать (« n = число \rightarrow прнт»)	Линейная последовательность основных операторов	D_{01} – задание числа D_{02} – вывод числа $\Delta = D_{01} + D_{02}$
1. Получить от ГСЧ и вывести случайное число (« n = ГСЧ \rightarrow прнт»)	Линейная последовательность основных операторов	D_{11} – порождение числа D_{01} – вывод числа $D_1 = D_{11} + D_{02}$ $\Delta = D_{11}$
2. Получить от ГСЧ и вывести последовательность случайных чисел; оценить интегральные параметры последовательности (« $\langle n \rangle \rightarrow n$ »)	Регулярные данные и действия – цикл по счетчику	$\Delta = D_{21}$ – организация повторения $D_2 = D_{21} + D_{11} + D_{02}$

Таблица 3.2 (продолжение)

За. Получить от ГСЧ и вывести две исходные последовательности случайных чисел, по заданному правилу сформировать из них результирующую последовательность; оценить интегральные параметры всех последовательностей («2х <n> → <n>»)	Накопление сумм (S) и количества по категориям (K) и подсчет интегральных показателей	$D_3 = D_{31} + D_{32} + D_{33}$ D_{31} – разделение чисел по признаку $D_{32} = D_2$ с добавлением счетчика N D_{33} – нахождение среднего $\Delta = D_{31} + D_{32} + D_{33}$
3б. Получить от ГСЧ и вывести последовательность случайных чисел, по заданному правилу разделить ее на две результирующие последовательности; оценить интегральные параметры всех последовательностей («<n> → 2х <n>»)	– " – " – " –	– " – " – " –

4. Получить от ГСЧ последовательность случайных чисел, наполнить ими одномерный массив; оценить интегральные параметры последовательности и выполнить комбинаторные преобразования с получением новой последовательности в результирующем массиве; организовать вывод обеих последовательностей из массивов (<<n> → n → <n>»)	Регулярные данные, действия и линейная структура – одномерный массив; использование общего индекса для цикла и массива	D_{41} – запись в элемент массива D_{42} – вывод элемента массива $D_{42} = D_3 + D_{41} + D_{42}$ $\Delta = D_{41} + D_{42}$
5а. Аналогично 3а - по схеме «2х<n> → <n>», с массивами		$D_5 = D_{42}$
5б. Аналогично 3б - по схеме «<n> → 2х<n>», с массивами		$D_{52} = D_{42}$
6. Варианты 5а, 5б с использованием подпрограмм	Выделение законченных комплексов действий и реализация их как процедур	D_{61} – оформление подпрограмм; D_{62} – оформление списка параметров; $\Delta = D_{61} + D_{62}$ $L_{62} = L_{51} + L_{61}$

Таблица 3.2 (окончание)

7. Варианты 5–6 с наполнением и обработкой двумерной матрицы	Регулярные данные, действия и линейная структура-двумерный массив	D_{71} – организация вложенных циклов; D_{72} – управление двумя индексами; $D_{73} = D_{51} + D_{71} + D_{72}$, $D_{74} = D_{52} + D_{71} + D_{72}$.
8. Получить от ГСЧ последовательность случайных чисел, заполнить ими последовательный числовой файл; извлечение последовательности из файла и ее обработка как в пп. 1-6; вывод последовательностей в файлы; просмотр файлов; программа строится с	Регулярные данные: а) в памяти и линейная структура - одномерный массив; б) на носителе - последовательный файл	D_{81} – описание последовательного файла; D_{82} – работа с последовательным числовым файлом; $D_8 = \Delta = D_{81} + D_{82}$
9. То же с описанием и использованием структурированного массива «числовая строка»		D_{91} – описание последовательного файла записями сложной структуры; D_{92} – обмен данными с файлом со сложной структурой данных; $D_9 = \Delta = D_{91} + D_{92}$

Обобщенные последовательности проблемных ситуаций для двух парадигм обучения - «от конкретного к общему» и от «от общего к конкретному», реализуется в рамках триады «Д-Г-Т», где буквы русского алфавита обозначают плоскости трехмерной модели основных классов проблемных ситуаций: Д - плоскость действия, Г - генетическая плоскость, Т - плоскость трудности; здесь

подстрочными индексами задаются классы проблем в этих плоскостях [80].

Предложенная А.М. Матюшкиным [80] трехмерная модель основных классов проблемных ситуаций, возникающих в регуляции и развитии действия, в том числе и действия по управлению – социальному управлению или управлению решением задачи при инфомоделировании на ЭВМ, использует три главных основания, три плоскости трехмерной модели классификации проблем. При этом рассогласование между прямой и обратной связью А.М. Матюшкиным было отнесено: а) к основным элементам действия; б) к различным этапам становления действия; 3) к различным степеням рассогласования в процессе становления и регуляции действия в связи с творческими возможностями субъекта действия. Названные аспекты отражены в виде трех измерений, в трех плоскостях объемной модели, на пересечении которых выделяются двадцать семь типов проблемных ситуаций, имеющие предметные и личностно-мотивационные различия:

Проблемные ситуации, описываемые плоскостью действия (Д-плоскость), могут служить для описания основных типов проблемных и определения их конкретного числа при обучении некоторой дисциплине, для определения основных шагов при обучении.

В соответствии с выделенными отношениями плоскость Д характеризует функциональное строение действия и включает его структурные компоненты: цель, определяющую предмет действия, способ действия и условия выполнения действия. В этой плоскости все возможные несоответствия, возникающие в проблемных ситуациях, могут быть описаны в соответствии с их предметным содержанием, представленным в модели в особенностях цели, выделяющей предмет действия и через выделенный предмет, в значительно степени определяющей условия и способ действия.

Три основных типа рассогласований в регуляции действия возможны в этой плоскости – на уровне цели, способа или условий действия [80], которые вызывают три типа проблемных ситуаций, определяемых специфическим характером между прямой и обратной связью. Рассмотрим три типичных примера трех типов рассогласования, характеризующиеся различным местом в структуре, которое неизвестное занимает в проблемной ситуации и которое совпадает с компонентами действия:

а) проблемные ситуации, в которых рассогласование имеет место на уровне предмета действия - теоретические ситуации (А.М. Матюшкин). В ПС этого типа требуется раскрыть новую закономерность, отношение

и т.п., необходимые для объяснения некоторого явления или для доказательства истинности определенного положения. Здесь неизвестное совпадает с целью действия: ...

б) проблемные ситуации с неизвестным способом действия создаются практическими задачами с заранее известной целью действий, составляющей вещь, состояние, процесс и т.п. Для выполнения этих заданий в проблемной ситуации необходимо найти новый способ достижения цели – например, найти новый метод решения сформулированной задачи, построить новый алгоритм, записать программу, реализующую алгоритм.

в) неизвестное составляет условия действия в широком круге ситуаций, возникающих при тренировках, закреплении действия, когда обучаемый усвоил принцип и способ выполнения действия. Особыми случаями таких ситуаций являются ситуации формирования навыков некоторых действий.

Генетическая плоскость (Г-плоскость) задает типы проблемных ситуаций, определяемые уровнями развития действия обучаемого, определяемыми уровнем психической регуляции, проявляемым в психологических особенностях прямой и обратной связи. В [80] предложено три относительных уровня, но на каждом уровне может быть введена и большая дробность, что увеличит и число возможных типов проблемных ситуаций.

Пересечение предметной и генетической плоскостей порождает девять возможных типов проблемных ситуаций, возникающих при открытии нового знания на основных этапах развития действия. При этом качественная характеристика каждого типа будет определяться местом главного фактора в структуре действия на некотором этапе его развития. Например:

- первый уровень как показатель относительно высокого уровня регуляции действия показывает возможность предвосхищения цели действия и рассогласования этой цели с реально достигаемым результатом;

- второй, более высокий уровень развития действия означает возможность предвосхищения не только цели, но и способа выполнения действия;

- на третьем уровне развития действия появляется возможность упреждающего анализа не только цели или способа, но и условий действия – здесь регуляционная основа изменяется качественно, обратная связь из внешней превращается во внутреннюю.

Генетическая плоскость характеризует степень развития регуляционной основы действия, составляющей содержание прямой связи (п/с). В условиях практической деятельности Г-плоскость при пересечении с Д-плоскостью позволяет определять последовательность проблемных ситуаций, ведущую от низкого уровня развития действия к более высокому.

Плоскость трудности ПС (Т-плоскость) характеризует степень рассогласования между ранее усвоенными и подлежащими усвоению знаниями в зависимости от творческих способностей обучаемого. Введение специальной Т-плоскости позволяет измерять степень проблемности учебных заданий, составляющей трудность проблемной ситуации для различных учащихся [80].

Пересечение Т-плоскости с Д-плоскостью и Г-плоскостью образует 27 классов проблемных ситуаций, возникающих в процессе развития действия. Каждая из этих ПС характеризуется ее местом в структуре действия, уровнем развития действия и измерена по степени рассогласования в процессе регуляции действия.

Т-плоскость позволяет производить микроанализ условий возникновения неизвестного и его выявления в ПС. Определение степени информативности ПС особенно важно в педагогической практике для целей индивидуализации обучения и для создания оптимальной системы обучения, адаптированной к различным возможностям обучаемых [80].

Предложенная А.М. Матюшкиным общая модель в случае обучения программному моделированию требует устранения принятых условностей, особенно применительно к этапам развития действия и к степенным трудности ПС; градация уровней и степеней трудности будет существенно большей, в частности, плоскость предметного строения действия должна соответствовать структуре конкретных действий при программном моделировании как управлении ресурсами ЭВМ.

В дальнейшем на базе общей модели классификации проблемных ситуаций, предложенной А.М. Матюшкиным, будет построена конкретная классификация проблемных ситуаций при изучении основ алгоритмизации и программирования.

Структурирование курса программирования по схеме проблемного обучения, его теоретической и особенно практической части, на наш взгляд, позволит индивидуализировать процесс обучения студентов, сделать его более управляемым и гибким, и. в конечном

счете, повысить эффективность подготовки будущих специалистов управления всех видов.

Гипотеза о подобии деятельности по управлению и деятельности по программному моделированию для одной предметной (профессиональной) области открывает направление для более глубокого и детального исследования мыслительных процессов, лежащих в основе этих видов деятельности, с целью выявления развивающего потенциала встречного обучения основам управления и основам программирования.

Глубокое структурирование курса программирования для студентов-непрограммистов, гибкое выстраивание его теоретической и особенно практической части, позволит индивидуализировать процесс обучения, сделает его более целенаправленным, управляемым и гибким, и через развивающее изучение основ программирования, через законченное решение задач с помощью программирования – алгоритмического или в других парадигмах, фактически даст студентам начальный опыт управления, что, в конечном счете, повысит эффективность подготовки будущих специалистов.

3.3. Анализ информационного моделирования как вида организационного управления. Управление и информационное моделирование – два вида конструктивно-логической деятельности над предметной областью

Ранее нами была выдвинута и обоснована гипотеза сближения и взаимопроникновения двух видов деятельности, и на этой основе предлагаются подходы к организации их взаимодействия с целью повышения эффективности управления.

Рассматривая некоторую предметную область и управление ею, будем считать что в конечном счете организационное управление и информационное моделирование над этой областью решают общую задачу - взять предметную область под эффективный контроль, управлять ею, но, взятая каждая по отдельности, они решают задачу лишь в определенной части - либо недостаточно строго, слабо формализовано, а значит неточно, как в случае оргуправления, либо не в полном объеме - как это делает инфомоделирование. Объединив эти научные направления, например, через разработку информационных продуктов и обучение универсальных специалистов -

информатиков-управленцев, можно преодолеть указанные недостатки, поднять управление на новый качественный уровень.

Аналогия между названными видами деятельности позволила выдвинуть гипотезу, что обучение программированию, информационному моделированию может использоваться как инструмент развития мышления, формирующий и развивающий многие операционные умения и навыки управленческой деятельности. Разумеется, управление и его разновидности, могут существенно отличаться от программирования в исполнительной практической части, но они во многом подобны в регуляционной, ориентировочной части.

Для эффективного сочетания и взаимодействия двух видов деятельности над единой предметной областью необходимо произвести их всесторонний психологический анализ, найти общее и выделить специфическое, сделать общее объектом целенаправленного психолого-педагогического воздействия на стадии обучения и подготовки специалистов.

Ранее, начиная с 60-х гг. прошлого века, о психологических особенностях программистов, о способностях к программированию и привлечении психологии к подготовке программистов писали как наиболее продвинутые программисты, так и профессиональные психологи.

Предлагается взглянуть на проблему шире:

- с одной стороны, оценить насколько психология необходима для программирования, в частности, и для информационного моделирования, в целом, как она помогает программисту и коллективу программистов в управлении собственным творчеством и коллективной работой;

- с другой стороны, как любому управленцу - менеджеру, педагогу и многим другим специалистам освоение методологии и инструментов инфомоделирования способно помочь в эффективном управлении своей предметной областью;

- целенаправленно действовать в направлении объединения как указанных видов деятельности, так и обучения этим видам деятельности.

На наш взгляд, методология и технологии инфомоделирования и психология управления с конца 50-х гг. прошлого века неуклонно сближаются, проникают друг в друга, и сегодня ситуация такова, что необходимым звеном, связывающим эти взаимодействующие научные направления, стимулирующим их имплицитное сближение, на наш взгляд,

может и должны стать психология и педагогика, точнее, педагогическая психология.

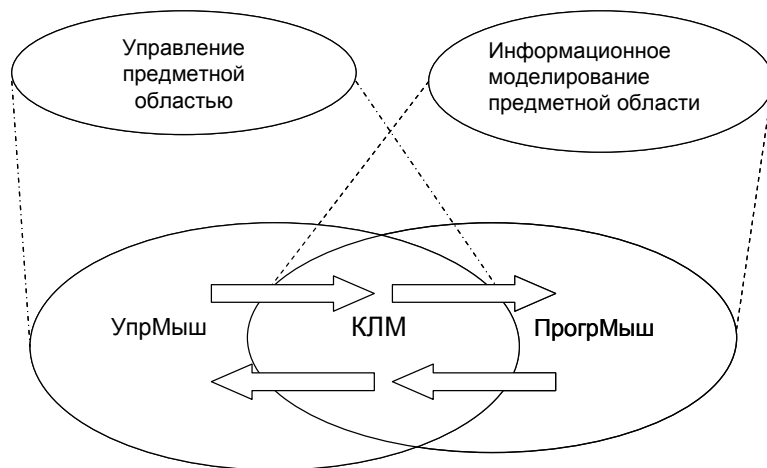


Рис. 3.1. Виды деятельности над предметной областью и опорные виды практического мышления:

УпрМыш – управленческое мышление менеджера-управленца предметной области;

ПрогрМыш (ИнфоМыш) – мышление программиста-информатика, моделирующего программно предметную область;

КЛМ – конструктивно-логическое мышление – КЛМ как пересечение и транзитный вид практического мышления при развитии одного вида мышления через обучение другому. Формирование и развитие КЛМ это предмет педагогической психологии.

На рисунке показано сближение двух видов деятельности до пересечения, и в перспективе - до слияния, отождествления.

В целом, как показано в выше, основные компоненты деятельности специалиста, занятого управлением в некоторой предметной области, подобны компонентам деятельности программиста, а, значит, подобны и компонентам учебной деятельности студента, обучающегося программированию.

Данная аналогия носит поверхностный, феноменологический характер, и требует более глубокой теоретической проработки и экспериментальной проверки, но позволяет выдвинуть гипотезу, что обучение программированию, информационному моделированию может использоваться как инструмент развития конструктивно-логического мышления, формирующий и развивающий многие операционные умения и навыки управленческой деятельности, формирование конструктивно-логической компетентности.

Как научные дисциплины организационное управление и информационное моделирование зародились в конце XIX - начале XX вв., но, если первая из них стала ответом на потребности нового этапа научно-технической революции в промышленности, то вторая - ответом на общий кризис в естествознании, в целом, и в математике, в частности - этим ответом стала конструктивная математика, в частности, теория алгоритмов и алгоритмические модели выдающихся математиков и логиков конструктивного направления Э.Поста, А. Тьюринга, А.А. Маркова.

На протяжении XX столетия указанные направления развивались, и продолжают развиваться параллельно, независимо, но неуклонно сближаясь. И, если первая научная дисциплина направлена на организации как социотехнические системы, то вторая - на программное и шире - информационное моделирование задач на ЭВМ, т.е. на технические системы, включая в себя и управление, выполняемое человеком-программистом или коллективом программистов-разработчиков.

В настоящее время, глубокому и всестороннему анализу психологии управления [36, 83-86] сегодня могут быть противопоставлены либо не слишком убедительные попытки исследования психологии программирования самими программистами [137] - неубедительные уже потому, что они не опираются на психологические методы изучения. Например, если учебно-методическая работа А.В. Карпова [36] выполнена профессионально в рамках теории деятельности и традиций фундаментальной отечественной школы психологии, то исследование психологических аспектов информатизации, выполненные психологами Московского университета [1, 2] носит скорее обзорный характер.

Следуя [36, 110], для обоснованного установления сходства управленческого мышления и мышления информатика, занимающего информационным моделированием некоторой предметной области, на наш взгляд, необходимо последовательно провести анализ содержания и

структуры деятельности по инфомоделированию предметной области с перечнем функций (когнитивные функции), выявить в деятельности программиста и информатика перцептивные и интегральные процессы (регулятивные, процессы принятия решения, коммуникативные, эмоционально-волевые, мотивации), для того, чтобы обоснованно сопоставить эту деятельность с деятельностью управленца, найти резервы для взаимодействия и взаимопроникновения.

Приведенный анализ управленческой деятельности позволяет установить соответствие между двумя обсуждаемыми видами конструктивно-логической деятельности - управлением и программированием, которое иллюстрируется следующей таблицей.

Таблица 3.3

Аналогии между видами конструктивно-логической деятельности

<i>Организационное, социальное и педагогическое управление</i>	<i>Информационное моделирование (программирование)</i>
Компоненты деятельности по управлению ситуацией: типовые последовательности, действия-, приемы, операции	Компоненты деятельности по программированию задачи: алгоритмы, алгоритмические конструкции, операторы, операции
Проблемная ситуация (С) и ее компоненты	Задача, решаемая на ЭВМ, и этапы/шаги ее решения
Ресурсы (Р) управления	Ресурсы ЭВМ: память, данные, процессорное время

Действия (Д) исполнителя по анализу и разрешению проблемной ситуации	Действия, процедуры и операции по преобразованию данных, задаваемые программистом или обучаемым в ходе программирования задачи
--	--

Из таблицы видно, что основные компоненты деятельности специалиста, занятого управлением в некоторой предметной области, аналогичны компонентам деятельности программиста, а значит и учебной деятельности студента, обучающегося программированию.

Программирование может рассматриваться как целенаправленная деятельность по управлению ресурсами вычислительной машины для решения задачи. Управление в своей операциональной части может рассматриваться как программирование – структурирование и организация обработки имеющихся ресурсов. Учебное программирование может рассматриваться как моделирование мыслительной деятельности по управлению в некоторой предметной области, к которой относится решаемая задача.

Исходя из выявленной аналогии, обучение программированию может использоваться как инструмент развития конструктивно-логического мышления, формирующий и развивающий ключевые компетенции (умения и навыки) управленческой деятельности. Разумеется, управление и его разновидности, могут существенно отличаться от программирования в исполнительно-практической части, но во многом подобны в регуляционной, ориентировочной части.

Три уровня управления – три подхода к развивающему обучению. Наша цель - выделить общие моменты в деятельности управленца и программиста, и использовать развивающее обучение программному моделированию не как обучение студентов основам управления, а как средство формирования КЛМ и, уже через него развитие УпрМыш. В основу такого обучения, на наш взгляд, целесообразно положить синтез как известных концепций в психологии компьютеризации, разрабатываемых в рамках смысловой теории мышления О.К. Тихомирова и его последователями - на факультете психологии Московского университета, а также подходов, предложенных

Предлагается выделять три аспекта - три уровня управления, и, соответственно, три подхода к развивающему обучению.

Любую ситуацию управления можно рассматривать в трех аспектах - временном, пространственном и структурном, следовательно, управление строится на этих уровнях, их сочетаниях. Тогда на такие же три уровня целесообразно разделять и КЛМ, а затем искать три канала воздействия на эти уровни. Такими каналами, на наш взгляд, могут служить три педагогические технологии программного моделирования задач:

1) представление задачи как последовательности действий во времени - алгоритмическая, последовательная, распределенная во времени (обучение программированию для ЭВМ на алгоритмических языках);

2) пространственная, где задача это программируемая плоская электронная таблица;

3) объектно-ориентированная декомпозиция задачи - разбиение задачи на совокупность взаимосвязанных и взаимодействующих объектов (объектно-ориентированное программирование - ООП).

Резерв повышения эффективности обучения заключается в более глубокой и детальной психолого-педагогической проработке процессов обучения и усвоения знаний при изучении программирования, с использованием второй и третьей технологий программного моделирования, а также с использованием концепции проблемного обучения, когда решение задачи организовано как поиск неизвестного, как обнаружение нового знания.

По результатам теоретического анализа проблем микро- и макроразвития мышления применительно к оперативному мышлению как виду практического мышления Д.Н.Завалишиной из ИП РАН [27-29] сформулированы принципы обучения применительно к профессиональному обучению («в первую очередь формированию практического мышления» - в частности, принцип единства специализации-лабилизации обучения, который «утверждает необходимость общего развития мышления как основы его широкой специализации и гибкости, недостаточность в качестве основы профессионального мастерства узкой специализации, связанной с формированием конкретных, специфичных для данной деятельности ЗУН и обеспечивающих их оперативных механизмов». Использование принципов управления при обучении инфомоделированию, как и принципов инфомоделирования при обучении основам управления, позволит существенно повысить эффективность обучения, как и

профессиональную компетентность управленца и программиста, их сотрудничества.

Этот же принцип ранее был положен нами в основу экспериментального исследования [48] на тему «Дидактические условия развития конструктивно-логического мышления студентов - будущих педагогов-математиков». В этом исследовании была теоретически и экспериментально исследована первая из технологий, и в сочетании с фундаментализацией процесса обучения был достигнут существенный педагогический эффект.

В обучении будущих специалистов сферы управления целесообразно использовать средства формального описания и анализа предметной области, программного моделирования управленческих ситуаций, синтеза управленческих решений. В [49-69] начата разработка подходов и методики такого обучения, теоретических и практических занятий по формированию конструктивно-логического мышления студентов-будущих управленцев.

Гипотеза о подобии деятельности по управлению и деятельности по программному моделированию для одной предметной (профессиональной) области открывает направление для более глубокого и детального исследования мыслительных процессов, лежащих в основе этих видов деятельности, с целью выявления развивающего потенциала встречного обучения основам управления и основам информационного моделирования.

Глубокое структурирование существующих учебных информационных дисциплин для студентов-непрограммистов, как и специальное изучение студентами-программистами основ психологии и организационного управления, гибкое выстраивание теоретической и особенно практической частей позволит индивидуализировать процесс обучения, сделает его более целенаправленным, управляемым и гибким, и через развивающее изучение основ инфомоделирования и психологии управления, через законченное решение задач инфомоделирования как задач управления, а задач управления как задач инфомоделирования с помощью программных систем, позволит обучающимся приобрести ценный начальный опыт современного управления, что, в конечном счете, повысит эффективность подготовки будущих специалистов.

3.4. Рефлексивный анализ управления и программирования. Подход к обучению студентов в рамках парадигмы персональной поддержки управленческой деятельности

Потребности развития общества и становления инновационной экономики требуют перестройки всей системы управления и ее отдельных звеньев в соответствии с гуманитарной и инновационной парадигмами [74, 75], что особенно важно в условиях всесторонней компьютеризации управления. Необходимы не только принципиальные изменения в профессиональной деятельности специалистов управления и информатиков-специалистов по компьютеризации и информационному моделированию в рамках парадигмы персональной поддержки управленческой деятельности, но и новые подходы к обучению студентов вуза, в их подготовке к этим видам деятельности.

Можно утверждать, что, в конечном счете, организационное управление и информационное моделирование решают общую задачу, а именно – взять предметную область под эффективный контроль, управлять ею. Но каждая в отдельности, они решают задачу лишь в определенной части – либо недостаточно строго, слабо формализовано, а значит неточно, как в случае управления, либо не до конца – как это делает инфомоделирование.

На наш взгляд, объединив эти научные направления в рамках субъектно-ориентированной концепции компьютеризации управленческой деятельности и учитывая особенности компьютеризации этого вида деятельности, выявленные В.Е. Лепским в [74, 75], можно преодолеть указанные недостатки, поднять управление на новый качественный уровень, во-первых через разработку информационных продуктов, а, во-вторых, путем комплексного развивающего обучения студентов, подготавливая их как современных управленцев и информатиков, а в перспективе, в идеале – как универсальных специалистов или управленцев-информатиков.

Предлагается взглянуть на проблему с еще одной позиции.

На наш взгляд, особого внимания заслуживают вопросы будущего взаимодействия двух категорий участников управления, которых необходимо учить совместной деятельности уже в вузе. Ниже рассматривается наиболее сложный интегрирующий коммуникативный процесс – процесс рефлексии, его место и роль в формировании профессиональных качеств у студентов.

Ранее утверждалось, что многие психологические компоненты деятельности специалиста, занятого управлением в социотехнической

системе, реализующего так называемый «технологический», инструментальный контур управления [36], подобны психологическим компонентам деятельности программиста «технической системы», а значит, подобны и психологическим компонентам обучения программированию задач для ЭВМ или инфомоделированию, в широком смысле. С другой стороны, и это очевидно – программист, тем более руководитель программного проекта и коллектива программистов, с необходимостью реализует и «человеческую» составляющую, экспрессивный контур управления – управляет своей предметной областью.

Это в определенном смысле поверхностное, внешнее подобие двух взаимодействующих видов деятельности требует углубленной проработки в рамках концепции персонализации управления и его субъектно-ориентированной компьютеризации [75], и, в частности, требуют дальнейшего изучения вопросы, относящиеся к рефлексивным процессам в управлении и инфомоделировании, а именно: а) вопрос роли и места в инфомоделировании ключевого метапроцесса любой сложноструктурированной деятельности – рефлексии; б) структура и механизмы рефлексивного взаимодействия участников процесса компьютеризации управления; в) дальнейшее изучение воздействия учебной деятельности по инфомоделированию как на основные структурные аспекты рефлексии будущего руководящего субъекта – кооперативный, коммуникативный, личностный и интеллектуальный, так и на формирование трех основных форм его рефлексии – ситуативную, ретроспективную и перспективную, зависящие от временных функций рефлексии [36, 159].

Если на первом этапе исследований [48] целью нашего психолого-педагогического исследования было выделить общее в деятельности управленца и программиста для использования обучения основам программирования в подготовке экономистов-управленцев, то в процессе исследования наши понимание проблемы и цель изменились – использовать встречное развивающее обучение программному моделированию и управлению двух категорий студентов для развития их профессиональных качеств. Для этого, с одной стороны, как ранее предложено в обучении программированию выделять три аспекта (три инструментальных уровня) управления ситуацией, такие же аспекты можно выделять и в обучении управлению. С другой стороны, в обучении будущих специалистов сферы управления предлагается шире использовать средства формального описания и анализа предметной области, ее

структурирования по различным измерениям, программного моделирования управленческих ситуаций, автоматизации синтеза управленческих решений – начата разработка средств и методологической основы совместного развивающего обучения студентов-будущих специалистов управления и студентов-информатиков. Так, например, со студентами специальности «Прикладная математика и информатика» и студентами-экономистами специальностей «Менеджмент» и «Математические методы в экономике» организованы три варианта занятий по решению учебных проблемных ситуаций: а) занятия, направленные на развитие персональной рефлексии управленцев и информатиков в рамках занятий по своей специальности; б) занятия, нацеленные на комплексную рефлекссию обеих сторон деятельности по управлению студентами каждой специальности по отдельности; в) совместные занятия студентов обеих специальностей по формированию навыков «групповой рефлексии разнопредметных специалистов» [36].

Субъектно-ориентированный подход к компьютеризации управленческой деятельности, персонализация в разработке автоматизированных информационных систем и систем управления [75], использование принципа рефлексии в подготовке студентов названных направлений и специальностей позволит сформировать и развивать в них способности к конструктивной деятельности, в целом, и к управлению в различных сферах, в частности. На современном этапе компьютеризации управления, подъем его на новый качественный уровень возможны лишь при совместных усилиях, тесной кооперации деятельности специалистов-разработчиков информационных систем и специалистов управления как на этапе создания систем, так и на этапах эксплуатации, сопровождения и доработки систем – к этому студентов необходимо готовить на разных стадиях профессиональной подготовки.

Сближение, взаимодействие и взаимопроникновение друг в друга названных видов профессиональной деятельности указывают направление перестройки всей системы вузовской подготовки соответствующих специалистов, что позволит обеспечить современный уровень их подготовки как специалистов современного, инновационного типа. Необходимы совместные усилия специалистов – психологов, педагогов, информатиков, менеджеров для дальнейшего исследования необходимых аспектов проблемы, для разработки методических и программных средств обучения студентов в рамках рассмотренной парадигмы, углубленного развития их способностей в указанных областях деятельности.

3.5. Три этапа и три парадигмы компьютеризации системы управления вуза (пример реализации предлагаемого подхода к обучению и управлению)

Потребности развития общества и модернизации системы профессионального образования требуют коренной реорганизации всей системы управления образованием и ее отдельных звеньев в соответствии с гуманитарной и инновационной парадигмами [74], что становится ключевым моментом в условиях всесторонней компьютеризации управления, когда необходима принципиальная перестройка в профессиональной деятельности как управленцев, так и специалистов по компьютеризации управленческой деятельности.

Создаваемая в Удмуртском госуниверситете Интегрированная информационно-аналитическая система (ИИАС) управления вузом (уже действуют ее первая версия, и внедряется вторая версия) прошла свой первый этап, и находится на втором этапе своего развития из трех, каждый из которых опирается на определенную парадигму компьютеризации управления:

1) технический этап – компьютеризация управления осуществлялась «внесубъектно, внедеятельностно ... в центр внимания ставились технологии сбора, анализа и обработки управленческой информации, а не конкретный руководитель, конкретное лицо, принимающее решение» [75], т.е. решались операционно-технические задачи управления, с минимальным учетом субъективного фактора управления;

2) социотехнический или организационно-технологический этап – субъекты управления адаптируются под созданную систему, обучаются компьютерным технологиям решения конкретных задач;

3) субъектно-ориентированный этап, содержанием которого должно стать преодоление «субъект-объектной» доминанты компьютеризации управленческой деятельности в пользу «субъект-субъектного» подхода – переход от поддержки управленческих решений к поддержке конкретных субъектов управленческой деятельности, с переходом синтезирующей парадигме «персональной поддержки управленческой деятельности» [75];

Первые два этапа с их подходами представляют собой проявление «субъект-объектного» подхода в компьютеризации управленческой деятельности, несут в себе все его признаки [75], а именно, слабо учитывают принципиальные особенности управленческой

деятельности, в целом, и управления в вузе, в частности.

Этап 1. Версия системы ИИАС-1, созданная специалистами-информатиками в рамках технического этапа на базе технологии и методологии фирмы ORACLE, реализовала обобщенную технологию решения управленческих проблем и адекватна лишь технологии информационных процессов, когда технологии компьютеризации управления рассматривались внедеятельностно и внесубъектно, когда в центр внимания поставлены технологии сбора, анализа и обработки исходной управленческой информации, и как следствие, «оставляла вне поля зрения все, что связано собственно с взаимодействиями и взаимодополнениями различными видами деятельности друг друга, процедуры кооперирования и синтеза различных видов деятельности, взаимодействия и отношения субъектов различных видов деятельности» [75].

Этап 2. Версия системы ИИАС-2 – это ИИАС-1, дополненная оргмероприятиями по перестройке процессов управления, а именно внедрением компонентов ИИАС на всех уровнях управления от нижнего – методист кафедры, специалист кадрового управления, бухгалтер и др., до самого верхнего уровня иерархии – главных специалистов всех направлений работы вуза и ректората.

Участники процесса компьютеризации были разделены на две категории-группы с подкатегориями: а) пользователи-управленцы, отвечающие – за всю предметную область, за отдельное направление, за участок направления; б) разработчик – руководитель проекта компьютерной системы, разработчик части компьютерной системы, специалист сопровождения системы и обучения пользователей-управленцев.

Для каждой категории разработаны технологии взаимодействия между участниками внутри группы, а также и между категориями и подкатегориями. Кроме того, начата реализация совместной работы различных субъектов управления, реализованы функции поддержки индивидуальной и групповой рефлексии.

Этап 3. В настоящее время начата разработка версии системы ИИАС-3, которая будет дополнена следующими масштабными управленческими функциями: а) контроль качества образовательного процесса в вузе на всех уровнях; б) оперативное управление учебным процессом, ведение он-лайн расписания и контроль графика учебного процесса; в) стыковка управляющей системы с создаваемой системой дистанционного электронного обучения.

В ИИАС-3 две парадигмы компьютеризации – «поддержка

управленческих решений» и «поддержка конкретных субъектов управленческой деятельности», будут дополнять друг друга, по отдельности они будут использоваться как частные стратегии решения отдельных задач, или как парадигма «персональной поддержки управленческой деятельности» [75].

Для достижения поставленной цели следует опираться на результаты психологического анализа, позволяющего выявить как общие принципиально важные особенности компьютеризации управленческой деятельности [75], так и особенности управления вузом. Для реализации субъектно-ориентированного подхода при проектировании следующей очереди системы ИИАС, на наш взгляд, необходимо: а) обеспечить и организовать в процессе управления групповую рефлексию как на стадии разработки системы, так и в процессе управления, для чего необходимо реализовать все виды оперативных коммуникаций в компьютерной сети как между субъектами внутри произвольной группы, так и между группами произвольного состава для трансформации деятельности лиц, принимающих решения (как субъектов деятельности) в деятельность совокупного субъекта групповой деятельности; например, в ИИАС-2 реализованы варианты внутрисетевого взаимодействия, но практика показала, что возникают межличностные психологические барьеры – пользователи не готовы взаимодействовать в произвольных группах, ограничиваясь парным взаимодействием; б) для поддержки индивидуальной и групповой рефлексии в ИИАС-3 необходимо реализовать многооконный индивидуально-настраиваемый интерфейс и графическую визуализацию числовых данных; в) необходимо организовать подготовку нового поколения пользователей-управленцев – подготовленных компьютеропользователей со сформированным конструктивно-логическим мышлением, способных осмысливать и формализовать бизнес-процессы управления разных уровней. Тогда станет возможным реализовать в рамках компьютеризированной технологии управления «взаимодействие и взаимодополнение различных видов деятельности, процедур кооперирования и синтеза различных видов деятельности, взаимодействие и отношения субъектов различных видов отношений» [75].

Субъектно-ориентированный подход, персонализация и всестороннее использование принципа рефлексии создадут предпосылки для более эффективной компьютеризации управленческой деятельности в вузе, создания следующих версий интегрированной информационно-аналитической системы – ИИАС-3, ИИАС-4 etc. В

этом случае можно ожидать от каждого участника процесса максимально персонализированной, ответственной и эффективной деятельности, что обеспечит наилучший результат управления, в целом.

3.6. Компетентности и компетентностный подход в подготовке информатиков и специалистов управления

Компетентностный подход [132; 144] впервые начал разрабатываться в Англии. Это был подход, который порождался и осмысливался не внутри образования, а был ответом на конкретный заказ профессиональной сферы. В США в сфере бизнеса в 70-х годах прошлого века стали использоваться понятия «компетенция» и «ключевые компетенции» в связи с проблемой определения качеств успешного профессионала.

Изначально компетенции стали противопоставляться специальным профессиональным знаниям и умениям. То есть начали рассматриваться как самостоятельные универсальные составляющие *любой* успешной профессиональной деятельности. Естественно, возник вопрос: можно ли научить компетенциям? Таким образом, проблематика компетенций попала в образование и со временем заняла в нем ведущее место.

Сфера образования, начиная с Я.А.Коменского, работала с основными единицами – знаниями, умениями и навыками (ЗУН). Профессиональная сфера работала с другими единицами – компетенциями (*профессиональные компетенции, ПК*). В этом смысле профессия дает ответ, какой компетентности должен быть человек или какова сфера его компетенции. Поэтому профессиональная сфера оперирует компетенциями, а образование – знаниями, умениями и навыками. И когда профессиональная сфера может точно на уровне заказа однозначно формулировать свои претензии к образованию, то задача образования заключается в том, как перекомпоновать знания, умения и навыки в определенные компетенции, которые требуются в профессиональной сфере.

Что такое компетентность? Устоявшегося определения для содержания понятия «компетенция» или «ключевая компетенция» до сих пор нет. Не существует и единой, принятой всеми классификации компетенций. Тем не менее, большинство авторов связывают компетентность с **эффективным выполнением какой-либо деятельности или действия.**

В Глоссарии терминов Европейского фонда образования (ЕФО, 1997) компетенция определяется как:

1. Способность делать что-либо хорошо или эффективно.
2. Соответствие требованиям, предъявляемым при устройстве на работу.
3. Способность выполнять особые трудовые функции.

То есть компетентность - это личностное свойство, характеристика, даваемая человеку в результате оценки эффективности/результативности его действий, направленных на разрешение определенного круга значимых для данного сообщества задач/проблем. Знания, навыки, способности, мотивы, ценности и убеждения рассматриваются как возможные составляющие компетентности, но сами по себе еще не делают человека компетентным.

В этом определении усматривается два подхода к содержанию понятия «компетенция». **Одни** исследователи делают **акцент** на компетенции как интегральном *личностном качестве человека* (характеристика человека), **другие** - на описании *составляющих его деятельности*, ее различных аспектов, которые и позволяют ему успешно справляться с решением проблем.

А.В. Хуторской [131] дает следующее определение понятия «компетенция»

Компетенция – отчужденное, заранее заданное социальное требование (норма) к образовательной подготовке ученика, необходимой для его эффективной продуктивной деятельности в определенной сфере.

Компетентность – владение, обладание учеником соответствующей компетенцией, включающее его личностное отношение к ней и предмету деятельности. Компетентность – уже состоявшееся качество личности (совокупность качеств) ученика и минимальный опыт деятельности в заданной сфере.

Таким образом, **компетентностный подход** предполагает не усвоение учеником отдельных друг от друга знаний и умений, а овладение ими в комплексе. В связи с этим меняется, точнее, по иному определяется система методов обучения. В основе отбора и конструирования методов обучения лежит структура соответствующих компетенций и функций, которые они выполняют в образовании.

Компетенции профессиональные следует отличать от образовательных компетенций, т.е. от тех, которые моделируют деятельность ученика для его полноценной жизни в будущем.

Образовательная компетенция – требование к образовательной подготовке, выраженное совокупностью взаимосвязанных смысловых ориентаций, знаний, умений, навыков и опыта деятельности ученика по отношению к определенному кругу объектов реальной действительности, необходимых для осуществления личностно и социально значимой продуктивной деятельности.

Компетенции для ученика – это образ его будущего, ориентир для освоения. Но в период обучения у него формируются те или иные составляющие этих «взрослых» компетенций, и чтобы не только готовиться к будущему, но и жить в настоящем, он осваивает эти компетенции с образовательной точки зрения. Образовательные компетенции относятся не ко всем видам деятельности, в которых участвует человек, например, взрослый специалист, а только к тем, которые включены в состав общеобразовательных областей и учебных предметов. Такие компетенции отражают предметно-деятельностную составляющую общего образования и призваны обеспечивать комплексное достижение его целей.

Ключевые компетенции

Сам термин «**ключевые компетенции**» указывает на то, что они являются «ключом», основанием для других, более конкретных и предметно ориентированных. Предполагается, что ключевые компетенции носят *надпрофессиональный и надпредметный характер* и необходимы в любой области деятельности.

В Оксфордско-Кембриджской образовательной программе «Ключевые компетенции 2000» указывается, что *ключевые компетенции* используются в повседневной жизни при осуществлении деятельности в области образования, на рабочем месте или при получении профессиональной подготовки.

В европейском проекте «Определение и отбор ключевых компетенций» (DeSeCo) *ключевые компетенции* определяются как важные «**во многих жизненных сферах и служащие залогом жизненного успеха и эффективного функционирования общества**».

Стратегия модернизации образования в РФ в основу обновленного содержания общего образования положила «ключевые компетентности»: «Основным результатом деятельности

образовательного учреждения должна стать не система знаний, умений и навыков *сама по себе*, а набор *заявленных государством* ключевых компетенций в интеллектуальной, общественно-политической, коммуникационной, информационной и прочих сферах».

Таким образом, *существенными признаками* ключевых компетенций являются:

1. Ключевые компетенции представляют собой различные универсальные ментальные средства, инструменты (способы, методы, приемы) достижения человеком значимых для него целей (результатов).

2. Ключевыми компетенциями в той или иной степени должен овладеть каждый член общества.

3. Ключевые компетенции позволяют человеку достигать результатов в не определенных, проблемных ситуациях. Они позволяют *самостоятельно* и в *сотрудничестве* с другими решать *проблемы*, то есть справляться с ситуациями, для разрешения которых *никогда нет полного комплекта наработанных средств*.

4. Определение и отбор ключевых компетенций осуществляется основными потребителями образовательных результатов на основе социологических исследований и общественного обсуждения и зависит от того, какие способности и качества человека являются ценными в данное время в данном обществе.

5. В современном западном обществе нормативную основу для отбора ключевых компетенций составляют базовые принципы прав человека, демократические ценности и цели, связанные с устойчивым развитием.

6. Компетенции проявляются и приобретаются человеком в деятельности, имеющей для него ценность.

Перечень ключевых компетенций основывается на главных целях общего образования, структурном представлении социального опыта и опыта личности, а также основных видах деятельности ученика, позволяющих ему овладевать социальным опытом, получать навыки жизни и практической деятельности в современном обществе.

Учебно-познавательные компетенции (УПК). Это совокупность компетенций ученика в сфере самостоятельной познавательной деятельности, включающей элементы логической, методологической, общеучебной деятельности. Сюда входят способы организации целеполагания, планирования, анализа, рефлексии, самооценки. По

отношению к изучаемым объектам ученик овладевает креативными навыками: добыванием знаний непосредственно из окружающей действительности, владением приемами учебно-познавательных проблем, действий в нестандартных ситуациях.

Учебно-познавательные компетенции:

- ставить цель и организовывать её достижение, уметь пояснить свою цель;
- организовывать планирование, анализ, рефлекссию, самооценку своей учебно-познавательной деятельности;
- задавать вопросы к наблюдаемым фактам, отыскивать причины явлений, обозначать свое понимание или непонимание по отношению к изучаемой проблеме;
- ставить познавательные задачи и выдвигать гипотезы; описывать результаты, формулировать выводы;
- выступать устно и письменно о результатах своего исследования с использованием компьютерных средств и технологий (текстовые и графические редакторы, презентации);
- иметь опыт восприятия картины мира.

Информационные компетенции (ИК). Навыки деятельности по отношению к информации в учебных предметах и образовательных областях, а также в окружающем мире. Владение современными средствами информации (телевизор, магнитофон, телефон, факс, компьютер, принтер, модем, копир и т.п.) и информационными технологиями (аудио- видеозапись, электронная почта, СМИ, Интернет). Поиск, анализ и отбор необходимой информации, ее преобразование, сохранение и передача.

Коммуникативные компетенции (КК). Знание языков, способов взаимодействия с окружающими и удаленными событиями и людьми; навыки работы в группе, коллективе, владение различными социальными ролями. Ученик должен уметь представить себя, написать письмо, анкету, заявление, задать вопрос, вести дискуссию и др. Для освоения этих компетенций в учебном процессе фиксируется необходимое и достаточное количество реальных объектов коммуникации и способов работы с ними для ученика каждой ступени обучения в рамках каждого изучаемого предмета или образовательной области.

Коммуникативные компетенции:

- уметь представить себя устно и письменно, написать анкету, заявление, резюме, письмо, поздравление;
- уметь представлять свой класс, школу, страну в ситуациях межкультурного общения, в режиме диалога культур, использовать для этого знание иностранного языка;
- владеть способами взаимодействия с окружающими и удаленными людьми и событиями; выступать с устным сообщением, уметь задать вопрос, корректно вести учебный диалог;
- владеть разными видами речевой деятельности (монолог, диалог, чтение, письмо), лингвистической и языковой компетенциями;
- владеть способами совместной деятельности в группе, приемами действий в ситуациях общения; умениями искать и находить компромиссы;
- иметь позитивные навыки общения в поликультурном, полиэтническом и многоконфессиональном обществе, основанные на знании исторических корней и традиций различных национальных общностей и социальных групп.

Социально-трудовые компетенции (СТК). Выполнение роли гражданина, наблюдателя, избирателя, представителя, потребителя, покупателя, клиента, производителя, члена семьи. Права и обязанности в вопросах экономики и права, в области профессионального самоопределения. В данные компетенции входят, например, умения анализировать ситуацию на рынке труда, действовать в соответствии с личной и общественной выгодой, владеть этикой трудовых и гражданских взаимоотношений.

Компетенции личностного самосовершенствования (КЛС) направлены на освоение способов физического, духовного и интеллектуального саморазвития, эмоциональной саморегуляции и самоподдержки. Ученик овладевает способами деятельности в собственных интересах и возможностях, что выражаются в его непрерывном самопознании, развитии необходимых современному человеку личностных качеств, формировании психологической грамотности, культуры мышления и поведения.

Использование компетентностной модели в образовании предполагает принципиальные изменения в организации учебного процесса, в управлении им, в деятельности учителей и преподавателей, в способах оценивания образовательных результатов обучающихся по

сравнению с учебным процессом, основанным на концепции «усвоения знаний».

Дело в том, что целый ряд школьных умений и знаний уже не принадлежит никакому профессиональному занятию. Примером такого экзотического вида школьных занятий может быть целый предмет черчение. Здесь остро необходима ревизия содержания образования. Во многих странах традиционные курсы трудового обучения и домоводства были заменены курсами «Технология и дизайн», «Предпринимательство». И это *все* - часть того обновления образования, которое проходит под лозунгами компетентностного подхода.

В *компетентностном подходе* перечень необходимых компетенций определяется в соответствии с запросами работодателей, требованиями со стороны академического сообщества и широкого общественного обсуждения на основе серьезных социологических исследований. Овладение различного рода компетенциями становится основной целью и результатами процесса обучения.

Основной ценностью становится не усвоение суммы сведений, а освоение учащимися таких умений, которые позволяли бы им определять свои цели, принимать решения и действовать в типичных и нестандартных ситуациях.

Позиции субъектов при компетентностном подходе

Позиция преподавателя Принципиально изменяется и позиция учителя, преподавателя. Он перестает быть вместе с учебником носителем «объективного знания», которое он пытается передать ученику. Его главной задачей становится мотивировать учащихся на проявление инициативы и самостоятельности. Он должен организовать самостоятельную деятельность учащихся, в которой каждый мог бы реализовать свои способности и интересы. Фактически он создает условия, развивающую среду, в которой становится возможным выработка каждым учащимся на уровне развития его интеллектуальных и прочих способностей определенных компетенций.

Развитие обучающегося

Меняется и смысл термина «развитие». Индивидуальное развитие каждого человека связано в первую очередь с приобретением умений, к которым у него уже есть предрасположенность (способность), а не с приобретением тематической информации, которая не только никогда

не понадобится в практической жизни, но и, по сути, не имеет никакого отношения к его индивидуальности.

Оценка уровня достижений ключевых компетенций

В целом освоенность тех или иных компетенций в учебном процессе можно оценивать как по результату разрешения проблемных ситуаций (в соответствии с заранее разработанными критериями успешности достижения этих результатов), так и в самом процессе их применения, использования, опять же по известным критериям.

Оценивают уровень владения конкретными компетенциями в таких процедурах, как *наблюдение за исполнением действий учащегося* в конкретных ситуациях, связанных с формированием, определенных компетенций (исследование, дискуссия, выступление и т.д.), письменного экзамена - написание эссе, а также и в других видах письменных экзаменов, решение тестовых заданий типа PISA.

Таким образом, компетентностный подход является усилением прикладного, практического характера образования всех уровней и форм (в том числе и предметного обучения). Это направление возникло из простых вопросов о том, какими результатами образования может воспользоваться школьник вне школы, выпускник вуза – начиная работать в своей профессиональной области. Ключевая мысль этого направления состоит в том, что для обеспечения «отдаленного эффекта» образования все, что изучается, должно быть включено в процесс употребления, использования. Особенно это касается теоретических знаний; которые должны перестать быть мертвым багажом и стать практическим средством объяснения явлений и решения практических ситуаций и проблем [132].

Выводы

Ранее нами было введены более корректные и более емкие понятия «конструктивно-логического мышления» (КЛМ) и «конструктивно-логическая деятельность» (КЛД), к которой отнесено информационное моделирование некоторой предметной области и, в частности, алгоритмизация и программирование задач для ЭВМ, а также управление некоторой предметной областью. Отсюда естественным образом родился связанный блок понятий и терминов - «конструктивно-логическая одаренность» (КЛО) у дошкольников, «конструктивно-логические способности» (КЛС) школьников и студентов, «конструктивно-логическая компетенция (КЛК) как

профессиональная компетенция для различных областей деятельности, как ключевая компетенция в обучении информатика и программиста, специалистов для различных областей организационного и социотехнического управления. Естественной видится и последовательность изучения и построения этапов формирования КЛК: 1) выявление одаренности КЛО; 2) развитие КЛО до способностей КЛС (от начальной школы до выпуска из школы); 3) формирование КЛК студента до начального профессионального уровня.

Опираясь на опыт исследований по развитию КЛМ у студентов-будущих преподавателей математики, в рамках конструктивно-логической компетенции предлагается выделять составляющие ее компетенции:

- совместно с заказчиком работы и аналитиком **составление технического задания** на информационный проект;
- знание **основ системного анализа и структурирования задач** на подзадачи;
- **алгоритмизации решения** отдельных подзадач, с объединением отдельных решений в решение поставленной задачи;
- **программирование** или кодирования алгоритма решения в терминах выбранного языка программирования;
- **отладка** программных модулей, реализующих подалгоритмы решения отдельных подзадач;
- **сборка** отлаженных модулей в модуль решения задачи и его отладка;
- **составление** технической **документации** на выполненный проект;
- **обучение конечных пользователей** работе с созданным информационным продуктом.

При обучении будущих специалистов для различных областей профессиональной деятельности отдельные компетенции из приведенного списка становятся ключевыми, и из них составляются в различные сочетания ключевых компетенций и степень их наполнения. Если для подготовки бакалавра выделяется существенно меньше учебного времени, то часть компетенций может быть исключена из списка, или уровень их освоения может быть заметно ниже. Если для будущих информатиков-технологов, многие тонкости современного программирования могут быть оставлены за кадром, но потребуют существенно больше внимания в качестве ключевых общекультурные, организационные и педагогические компетенции.

Заключение

В первой и второй главах изложены результаты исследовательской экспериментальной работа по обучению программированию и развитию конструктивно-логического мышления студентов-будущих преподавателей математики и информатики. Основной задачей было, с одной стороны, показать эффективность разработанных дидактических условий, а, с другой стороны, доказать, что созданная дидактическая модель и разработанная методика обучения способствует повышению успешности обучения программированию, и тем самым обеспечивает существенное развитие конструктивно-логического мышления студентов.

Если в работах [48-68] были исследованы различные аспекты диагностики и развития конструктивно-логического мышления у студентов, детально изучены студенты-непрограммисты, а также исследованы предпосылки успешного развития этого вида практического мышления у дошкольников, то для законченности исследования, на наш взгляд, логично обратить пристальное внимание на определяющие этапы развития личности учащегося – начальную и среднюю школу, причем, равное внимание должно быть уделено двум категориям учащихся средних учебных заведений – обычным учащимся, неориентированным специально на информатику и программирование, а также школьникам-участникам кружкового движения по информатике, дополнительного образования и олимпиад по программированию – эта работа в настоящее время проводится нами совместно с В.В. Пупышевым [69].

В результате многоэтапного исследования будут созданы предпосылки для построения законченной дидактической системы формирования конструктивно-логической компетентности будущих специалистов в области информатики и управления – от выявления одаренности дошкольника и младшего школьника, через развитие способностей школьника в среднем и старшем звене, до формирования начальных уровней профессионализма в информатике и управлении у выпускников высших учебных заведений.

Дидактическая система по В.П. Беспалько, как и по В.П. Симонову, включает следующие компоненты, которые конкретизируются в рамках проводимого исследования:

1) **цели образования** (в широком смысле): создание дидактических условий развития КЛЮ дошкольника до максимально возможного индивидуального уровня;

2) **содержание учебно-воспитательного процесса**: выявление, оценка и развитие КЛЮ детей в дошкольный период и в начальной школе; выявление, оценка и развитие КЛС школьников среднего звена на уроках информатики и в структурах дополнительного образования, а для высокого уровня КЛС в рамках олимпиадного движения по информатике и программированию;

3) **учащиеся**: дошкольники в ДДУ, школьники всех звеньев, студенты профессиональных учебных заведений;

4) **сам учитель**: родители, воспитатели ДДУ, учителя и учителя-предметники школ, преподаватели вузов, специалисты-профессионалы из различных областей;

5) **дидактические процессы**: непосредственное или опосредованное педагогическое воздействие на мышление; проявление и вызревание КЛЮ в рамках воспитательных программ ДДУ и дошкольного дополнительного образования; трансформация поздно выявленной КЛЮ в КЛС низкого уровня в рамках общеобразовательных дисциплин, а созревшей КЛЮ в КЛС среднего или высокого уровня – в рамках кружков, дополнительных занятий по информатике, конструированию и т.п.;

6) **организационные формы и результаты** на различных этапах формировании КЛК существенно различаются.

Сложность поставленной задачи состоит еще и в том, что:

а) субъекты образовательного процесса существенно различаются по возрасту и социальному статусу, принадлежат к разным педагогическим системам (ПС);

б) в рамках общеобразовательных ПС обучаемые получают основное образование, а целенаправленное развитие КЛМ и формирование КЛК необходимо встроить в эти системы, дополнить их.

Тем не менее, целесообразно и в дальнейших исследованиях рассматривать именно приведенный состав создаваемой дидактической системы, поскольку ее перечисленные компоненты, представленные во взаимосвязи, достаточно полно характеризуют мотивационно-целевой, содержательный, процессуальный и контрольно-оценочный этапы образовательного процесса на всех стадиях проводимого педагогического эксперимента.

Список использованной литературы

1. Бабаева Ю.Д., Войскунский А.Е. Одаренный ребенок за компьютером. - М.: Сканрус, 2003. - 336 с.
2. Бабаева Ю.Д., Войскунский А.Е. Психологические аспекты одаренности детей и подростков в применении информационных технологий. Тезисы 2-й Российской конференции по экологической психологии (Москва, 12-14 апреля 2000 г.). - М.: Экопсицентр РОСС, 2000. - С.246-248.
3. Белая О.А., Новиков Б.А., Одинцов И.О. Психология программирования: человеко-машинный аспект информационных технологий/2-я Откр.Всеросс. конф. «Преподавание ИТ в России», 2004.
4. Бельтюков А.П. Дедуктивный синтез алгоритмов с учетом вычислительных ресурсов. Спец. 05.13.17 – Теоретические основы информатики. Автореф. дис. ... д-ра физ.-мат.наук. -СПбГУ:УдГУ, 1993. - 20 с.
5. Бельтюков А.П., Копотов С.Л., Купчинаус С.Ю. Конструктивно-логическое мышление в основе обучения математике и программированию // Сб. «Инновационные процессы в сфере образования и проблемы повышения качества подготовки специалистов», Сб. мат. междунар.науч.-практ. конф. 30-31 марта 2005. Т.2. - Ижевск. «Удм.ун-т». 2005. - С.69-75.
6. Беспалько В.П. Основы теории педагогических систем. Проблемы и методы психолого-педагогического обеспечения технических обучающих систем. - Воронеж: Изд-во Воронежского ун-та. 1977. - 304 с.
7. Беспалько В.П., Татур Ю.Г. Системно-методическое обеспечение учебно-воспитательного процесса подготовки специалистов: Учеб.-метод.пособие. - М.: ВШ, 1989. - 144 с.
8. Бордовский Г.А. и др. Проблемы педагогики информационного общества и основы педагогической информатики. В сб. Дидактическое основы компьютерного обучения. Межвуз.сб. науч.трудов. - Л.: ЛГПИ, 1989. - С.3-32.
9. Брушлинский А.В. Психология мышления и проблемное обучение. - М.: Знание, 1983. - 96 с.
10. Бурков В.Н. Коргин Н.А., Новиков Д.А. Введение в теорию управления организационными системами: Учебник. – М.: Книжный дом «ЛИБРОКОМ», 2009. 264 с.

11. Войскунский А.Е. Психология компьютеризации и оппонентные дисциплины / Современная психология мышления: смысл в познании. Тез. докл. науч. конф., поев. 75-летию со дня рожд. выд. отеч. Психолога О.К.Тихомирова (1933-2001), Москва 17-18 окт. 2008. М.: Смысл, 2008. С.227-229.
12. Выготский Л.С. Воображение и творчество в детском возрасте. - СПб: Изд. «Союз», 1997. - 92 с.
13. Гейн А.Г. Изучение информационного моделирования как средство реализации межпредметных связей информатики с дисциплинами естественнонаучного цикла: Дис. ... д-ра пед. наук. Институт общего среднего образования РАО (ИОСО РАО). - 307 с.
14. Гершунский Б.С. Философия образования для XXI века (В поисках практико-ориентировочных образовательных концепций). – М.: Совершенство, 1998.- 608 с.
15. Громов Г.Р. Национальные информационные ресурсы. Проблемы промышленного использования. - М.:Наука, 1984. – 264 с.
16. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. - М.: Мир, 1981. - 368 с.
17. Давыдов В.В. Проблемы развивающего обучения: Опыт теоретического и экспериментального психологического исследования. - М.: Педагогика, 1986. - 240 с.
18. Дейкстра Э. Алгоритмы + структуры данных = программа. - М.: Мир, 1985. - 350 с.
19. Дидактические основы компьютерного обучения. Межвуз. сборник научных трудов. - Л.: ЛГПИ. 1989. - 202 с.
20. Долныкова А.А., Чудова Н.В. Психологические особенности суперпрограммистов // Психол.журнал. т.18. №1. 1997. - С.113-121.
21. Дружинин В.Н. Психологическая диагностика способностей: теоретические основы. Часть II (продолжение). - Саратов: Изд-во Саратовского ун-та, 1990. - 160 с.
22. Дубова Н. Математика и ИТ // Открытые системы. 2004. № 10.
23. Ершов А.П. Избранные труды. - Новосибирск: ВО "Наука". Сибирская издательская фирма, 1994. - 416 с.
24. Ершов А.П. Компьютеризация школы и математическое образование // Информатика и образование. №№5-6, 1992. - С.3-12.
25. Ершов А.П. Программирование – вторая грамотность. В кн. Избранные труды. - Новосибирск: ВО Наука, 1994. - 416 с.
26. Ершов А.П. Человек и машина. Человеческий фактор в программировании. - М.: Знание, 1985. 32 с. (Новое в жизни, науке, технике. Сер. "Математика, кибернетика"; № 4).

27. Завалишина Д.Н. Практическое мышление: специфика и проблемы развития. - М.: Изд-во «Институт психологии РАН», 2005. 376 с.

28. Завалишина Д.Н. Психологический анализ оперативного мышления (экспериментально-теоретическое исследование). - М.: Наука, 1985. - 222 с.

29. Завалишина Д.Н., Пушкин В.Н. О механизмах оперативного мышления // Вопросы психологии. №3, 1964. - С.87-100.

30. Зинченко В.П. Гуманитарные проблемы информатики // Вопросы философии. 1986. № 9.

31. Изучение основ информатики и вычислительной техники: Пособие для учителя / А.В.Авербух и др. - М.: Просвещение, 1992. - 302 с.

32. Информатика: Энциклопедический словарь для начинающих / Сост. Д.А.Поспелов. - М.: Педагогика-Пресс, 1994. - 352 с.

33. Искусственный интеллект и психология / Под ред. О.К.Тихомирова. - М.: Изд-во МГУ, 1976.

34. Калашникова М.Б., Регуш Л.А. Психологические основы компьютеризации обучения. В сб. Дидактические основы компьютерного обучения. Межвуз.сб. науч.трудов. - Л.: ЛГПИ, 1989. - С.33-44.

35. Каптелин В.Н. Психологические проблемы формирования компьютерной грамотности школьников // Вопр.психологии. № 5, 1986. -С.54-65.

36. Карпов А.В. Психология менеджмента: Учеб. Пособие. - М.: Гардарики, 2006. 584 с.

37. Касаткин В. Н. Информация, алгоритмы, ЭВМ: Пособие для учителя. - М.: Просвещение, 1991. - 192 с.

38. Кастельс М. Информационная эпоха: экономика, общество и культура. - М.: Гос. Ун-т Высшая школа экономики, 2000. - 608 с.

39. Кларин М.В. Инновации в обучении: метафоры и модели: анализ зарубежного опыта. - М.: Наука, 1997. - 223 с.

40. Кларин М.В. Педагогическая технология в учебном процессе. Анализ зарубежного опыта. - М.: Знание, 1989. - 80 с.

41. Клейман Глен М. Школы будущего: компьютеры в процессе обучения. Пер с англ. М., 1987.

42. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиции математического моделирования. Под. ред. А.А.Самарского. - М. 1968.

43. Копотев С.Л. Педагогическая оценка личности учащегося. Метод.указания. - Устинов: Удмуртский ун-т, 1986. - 42 с.

44. Котенко В.В. Рефлексивная задача как средство повышения обучаемости школьников в процессе изучения базового курса информатики: Дис. ... канд. пед. наук. Омский государственный педагогический университет (ОмГПУ). - 168 с.

45. Кузьмина Н.В. Профессионализм деятельности преподавателя и мастера производственного обучения профтехучилища. - М.: Высшая школа, 1990. - 167 с.

46. Кузьмина Н.В. Профессионализм личности преподавателя и мастера производственного обучения. - М.: Высшая школа, 1990. - 119 с.

47. Купчинаус С.Ю. Дидактические условия развития конструктивно-логического мышления при обучении студентов-будущих преподавателей математики: Автореф.дис. ... канд.пед.наук. Удмуртский госуниверситет (УдГУ), 2006. - 20 с.

48. Купчинаус С.Ю. Дидактические условия развития конструктивно-логического мышления при обучении студентов-будущих преподавателей математики: Дис. ... канд.пед.наук. Удмуртский госуниверситет (УдГУ), 2006. - 181 с.

49. Купчинаус С.Ю. Дидактические условия развития конструктивно-логического мышления при обучении студентов программированию // Синергетика в междисциплинарном подходе современной психологии: Сб. науч. тр. Под ред. Р.Х.Тугушева. - Саратов: Изд-во Сарат. ун-та, 2005. - С.91-93.

50. Купчинаус С.Ю. Программное моделирование и развитие практического мышления студентов / В сб. Труды II Всеросс. научной конф. Технологии информатизации профессиональной деятельности ТИПД-2008. Ижевск: РХД, 2008. С. 263-271.

51. Купчинаус С.Ю. Программное моделирование как инструмент развития практического мышления / Современная психология мышления: смысл в познании. Тез. докл. науч. конф. М.: Смысл, 2008. С.280-282.

52. Купчинаус С.Ю. Роль дидактических принципов при обучении студентов программированию // Сб. науч.тез. 12-й Международной конференции «Математика. Компьютер. Образование». - Москва-Ижевск: РХД, 2005. - С.315.

53. Купчинаус С.Ю., О дидактической системе формирования конструктивно-логической компетентности: от выявления одаренности до начальных уровней профессионализма в информатике / В сб. Труды

III Всеросс. научной конф. Технологии информатизации профессиональной деятельности ТИПД-2011. Ижевск: «Удмуртский университет», 2011. С. 52.

54. Купчинаус С.Ю. Обучение программному моделированию как инструмент развития мышления будущего специалиста // Сб. «Мат-лы V Международ. Науч.-практ.конф. «Личностно-развивающее профессиональное образование» Ч.1». - Екатеринбург: РГППУ, 2005. - С.183-186.

55. Купчинаус С.Ю. О психолого-педагогических аспектах обучения программированию // Сб. «Тезисы докладов 4-й Росс. Универ.-академ. Науч.-практ.конф.», ч.6. - Ижевск: Изд-во Удм.ун-та, 1999.

56. Купчинаус С.Ю. Психолого-педагогические аспекты оценивания успешности обучения программированию студентов вуза. Вып. квалиф. работа. - Ижевск: УдГУ, 1999. - 78 с. (рукопись).

57. Купчинаус С.Ю. Обучение программному моделированию как инструмент развития мышления будущего специалиста // Личностно-развивающее профессиональное образование / Сб. «Материалы V Международ.науч.-практ.-конф.», 4.1. - Екатеринбург: РГППУ, 2005. - С. 183-186.

58. Купчинаус С. Ю. Субъектно-ориентированный подход в совместном обучении будущих управленцев и информатиков с использованием принципа рефлексии // Рефлексивные процессы и управление : сб. материалов VII Междунар. симп., 15-16 окт. 2009 г., М. / под ред. В. Е. Лепского. - М. : Когито-Центр, 2009. - С. 121-124.

59. Купчинаус С. Ю. Подход к обучению студентов в рамках парадигмы персональной поддержки управленческой деятельности // Психолого-педагогические проблемы одаренности: теория и практика : материалы VI Междунар. конф. В 2 т. / под ред. Л. И. Ларионовой. - Иркутск, 2009. - Т. 2. - С. 291-299.

60. Купчинаус С. Ю. Обучение студентов в рамках субъектно-ориентированной парадигмы компьютеризации управления // IV Международная научно-практическая конференция "Современные информационные технологии и ИТ-образование" : сб. тр. / под ред. В. А. Сухомлина. - М., 2009. - С. 177-184.

61. Купчинаус С. Ю. Комплексный подход к развивающему обучению студентов - будущих информатиков и специалистов управления // Информационная среда вуза XXI века : материалы III Междунар. науч.-практ. конф. (21-25 сент. 2009 г.) / отв. ред. Н. С. Рузанова. - Петрозаводск, 2009. - С. 140-143.

62. Купчинаус С. Ю. Программное моделирование и развитие практического мышления студентов // Технологии информатизации профессиональной деятельности [Электронный ресурс] : 2 всерос. науч. конф. с междунар. участием, Ижевск, 2008 / ГОУВПО "Удмурт. гос. ун-т", ФИТиВТ. - Ижевск : БОН АНЦА, 2008. - Ч. 1. - С. 263-271.

63. Купчинаус С. Ю. Программное моделирование и развитие мышления студентов // Девятая российская университетско-академическая научно-практическая конференция : материалы конф. / ГОУВПО "Удмурт. гос. ун-т", Естест.-гуманит. науч.-образов. комплекс (ЕГНОК). ; отв. ред. Н. И. Леонов. - Ижевск, 2008. - С. 137-139.

64. Купчинаус С. Ю. Компетенция в программировании как показатель развития конструктивно-логического мышления учащегося // Компетентностный подход в образовании : материалы и тез. Регион. науч.-практ. конф., 9 янв. 2006 г. / Упр. образования администрации г. Ижевска, Отд. упр. образования Администрации Устин. р-на г. Ижевска, МОУ Лицей № 41 [и др.]. - Ижевск : [б. и.], 2006. - С. 108-111.

65. Купчинаус С. Ю. Развитие конструктивно-логического мышления при обучении информационному моделированию // Парадигмы образования : материалы науч. сес., 26 апр. 2006 г. / Удмурт. гос. ун-т, Ин-т педагогики, психологии и социал. технологий. - Ижевск : [б. и.], 2006. - Т. 1. - С. 145-146.

66. Купчинаус С.Ю., Анисимов А.Е. Программирование. Ч. 1. Введение в алгоритмизацию задачи программирование на языке Паскаль. Учебное пособие. – Ижевск: УдГУ, 1994, 2004. - 80 с.

67. Купчинаус С. Ю. и др. Конструктивно-логическое мышление в основе обучения математике и программированию // Инновационные процессы в сфере образования и проблемы повышения качества подготовки специалистов : материалы междунар. науч.-метод. конф., Ижевск, 30-31 марта 2005 г. / под ред.: А. А. Баранова, Г. С. Трофимовой ; УдГУ, Учеб.-метод. совет. - Ижевск, 2005. - Т.2. - С. 75-79.

68. Купчинаус С.Ю., Кондак Е.С. / В сб. Труды II Всеросс. научной конф. Технологии информатизации профессиональной деятельности ТИПД-2008. Ижевск: РХД, 2008. С. 54.

69. Купчинаус С.Ю., Пупышев В.В. Конструктивно-логические составляющие будущей компетентности информатика / В сб. Труды II Всеросс. научной конф. Технологии информатизации профессиональной деятельности ТИПД-2008. Ижевск: РХД, 2008.

70. Купчинаус С.Ю., Сполохова М.В. / В сб. Труды II Всеросс. научной конф. Технологии информатизации профессиональной деятельности ТИПД-2008. Ижевск: РХД, 2008. С. 55.

71. Ланда Л.Н. Алгоритмизация в обучении. - М.: Просвещение, 1966. - 524 с.

72. Лапчик М.П. и др. Методика преподавания информатики: учеб. пособие для студ. пед. вузов. - М.: Академия, 2001. - 624 с.

73. Левченко И.В. Роль дидактических принципов в повышении эффективности алгоритмической подготовки школьников. Тезисы Междунар. конференции «Информационные технологии в образовании» ИТО-2003. - М.: ИТО, 2003.

74. Лепский В.Е. Гуманитарные технологии информатизации общества (на примере компьютеризации управленческой деятельности) // Информационное общество, вып. 6, 2001.

75. Лепский В.Е. Субъектно-ориентированная концепция компьютеризации управленческой деятельности. - М.: ИП РАН, МГУ им. М.В. Ломоносова, 2000. - 42 с.

76. Лернер И.Я. Дидактические основы методов обучения. - М.: Педагогика, 1981. - 186 с.

77. Лернер И.Я. Процесс обучения и его закономерности. - М.: Знание, 1980. - 96 с.

78. Ломов Б.Ф. Научно-технический прогресс и средства умственного развития человека // Психологический журнал, № 6, Т.6, 1985.

79. Макаренков Ю.А., Столяр А.А. Что такое алгоритм? Беседы со - старшеклассником. - Мн.: Нар.асвета, 1989. - 127 с.

80. Матюшкин А.М. Проблемные ситуации в мышлении и обучении. - М.: Педагогика, 1972. 168 с.

81. Машбиц Е. И. Психологические основы управления учебной деятельностью. - Киев: Высшая школа, 1987. - 223 с.

82. Машбиц Е.И. Психолого-педагогические проблемы компьютеризации обучения: (Педагогическая наука - реформе школы). - М.: Педагогика, 1988. - 192 с.

83. Менеджмент: Тесты, задачи, ситуации, деловые игры. Практикум: Учеб. Пособие /Беляцкий Н.П. и др. - Мн.: Книжный Дом, 2005. 224 с.

84. Менеджмент организации. Учеб. Пособие. Румянцева З.П. и др. - М.: ИНФРА-М, 1997.432 с.

85. Мескон М. и др. Основы менеджмента. - М.: Дело, 2004. 720 с.

86. П.Ревская Н.Е. Психология менеджмента. Конспект лекций. - СПб: Альфа, 2001.240 с.

87. Методы системного педагогического исследования. Уч.пособие. - Л.: Изд-во ЛГУ, 1980. - 172 с.

88. Мирошниченко И.Л. Приоритетно-логическое структурирование учебной информации. Автореф. дис. ... канд. пед. Наук. Глазовский пед.ин-т – Ижевск, 2005. - 23 с.

89. Моисеев Н.Н. Алгоритмы развития. - М. 1987.

90. Моляко В.А. Психологическая система тренинга конструктивного мышления // Вопросы психологии. № 5, 2000. - С.136-141.

91. Непейвода Н.Н. Какая математика нужна информатикам? // Открытые системы. №9, 2005.

92. Никитина В.Г. Развитие алгоритмического и логического мышления школьников // I-е городские Прохоровские чтения учителей математики и информатики. Якутск, 2001.

93. Николов Р. Сендова Е. Начала информатики. Язык Лого. - М.: Наука. Гл.ред.физ.-мат.лит. 1990. - 176 с.

94. Новик И. Б. О моделировании сложных систем (философский очерк). - М.: Мысль, 1965. - 335 с.

95. Новикова Т.А. Формирование основ теоретического мышления у младшего подростка в процессе естественнонаучного образования. Автореф. дис. ...канд.пед.наук. - Ижевск. 2004. - 20 с.

96. Новикова Т.А. Формирование основ теоретического мышления у младшего подростка в процессе естественнонаучного образования. Дис. ...канд.пед.наук. - Ижевск, 2004. - 152 с.

97. Овчинникова Т.Н. Личность и мышление ребенка: диагностика и коррекция. 2-е издание. - М.: Академ.проект; Екатеринбург: Деловая книга, 2000. - 208 с.

98. Основы дидактики. П/р проф.Б.П.Есипова. - М.: Просвещение, 1967. - 472 с.

99. Педагогика и психология высшей школы. - Р.-н-Д.: Феникс, 1998. - 544 с.

100. Петров П.К. Система подготовки будущих специалистов физической культуры в условиях информатизации образования : Дис. ... д-ра пед. наук : 13.00.01, 13.00.04 Ижевск, 2003ю – 406 с.

101. Петрова Ю.А. Дифференцированный подход при обучении объектно-ориентированному программированию в старшей школе : Автореф. дис. ... канд. пед. наук. Рос. гос. пед. ун-т им. А. И. Герцена.- СПб, 2002.

102. Поспелов Г.С. Информатика и НТП // Вопросы философии, № 9 1986.
103. Преподавание информатики как комплексная проблема./ Материалы раб.межвуз.совещания. 1.02-3.02.2005. УдГУ. Ижевск, 2005.
104. Программа курса основ информатики и вычислительной техники 1Х-Х классы (проект) /Под ред. А.П.Ершова. М.,1986.
105. Пушкин В.Н. Оперативное мышление в больших системах. – М.-Л.: Энергия, 1965. – 375 с.
106. Пушкин В.Н. Психология и кибернетика. – М.: Педагогика, 1971. – 232 с.
107. Реан А.А. Психология педагогической деятельности (проблемный анализ): Учеб.пособ. - Ижевск: Изд-во Удм.ун-та, 1994. - 83с.
108. Рожина И.В. Обучение учащихся объектно-ориентированному программированию и технологии визуального проектирования в базовом курсе информатики: Дис. ... канд. пед. наук. Уральский государственный педагогический университет (УрГПУ). - 176 с.
109. Российская педагогическая энциклопедия. В 2-х томах. - М.: Большая российская энциклопедия, 1993. – 608 с.
110. Рубинштейн С.Л. О мышлении и путях его исследования. - М., 1958.
111. Румянцев Д., Монастырский Л. Путь программиста. Опыт создания личности программиста. - М.: Изд.дом Инфра-М, 2000. – 864 с.
112. Русакова О.Л. Информатика (уроки развития). - Пермь: Изд-во Пермского ун-та, 1998. - 80 с.
113. Столяр А.А. Как математика ум в порядок приводит. - Мн.: Выш.школа, 1991. - 207 с.
114. Столяр А.А. Педагогика математики: Учеб. пособие для физ.-мат.фак. пед.ин-тов. - Мн.: Выш.шк, 1986. - 414 с.
115. Суходольский Г.В. Основы психологической теории деятельности. - Л. 1988.
116. Талызина Н.Ф. Управление процессом усвоения знаний (психолого-педагогические основы). - М.: Изд-во Моск.ун-та. 1984. 344 с.
117. Талызина Н.Ф., Карпов Ю.В. Педагогическая психология: психология интеллекта: Учеб.-метод.пособие. - М.: Изд-во Моск.ун-та. 1987. - 63 с.
118. Теплов Б.М. Ум полководца. – М.: Педагогика, 1990. – 208 с.

119. Терехина А.Ю. Структура знаний о языке программирования // Вопросы психологии. №4, 1988. - С.137-142.
120. Тихомиров О.К. Информатика и новые проблемы психологической науки // Вопросы философии. № 7, 1986.
121. Тихомиров О.К. Стратегия и тактика компьютеризации // Вестник высшей школы. № 3, 1988.
122. Тихомиров О.К. Структура мыслительной деятельности человека (опыт теоретического и экспериментального исследования). - М.: Изд-во МГУ, 1969. - 304 с.
123. Тихомиров О.К., Бабанин Л.Н. ЭВМ и новые проблемы психологии. Учеб.пособие для слушателей ФПК. - М.: Изд. МГУ, 1986. - 204 с.
124. Тростников В.Н. Конструктивные процессы в математике (философский аспект). - М.: Наука, 1975. - 255 с.
125. Трофимова Г.С. Основы педагогической коммуникативной компетентности: учеб.пособие. – Ижевск: Изд-во Удм. Ун-та. 1994. 76 с.
126. Тюрин Е.П. Объективные критерии сложности конкретно-практических задач по предметам математического и естественнонаучного циклов: Дис. ... канд. пед. наук. Институт теории образования и педагогики РАО (ИТОиП РАО). - 112 с.
127. Умрихин В.В. Предисловие / Теплов Б.М. «Ум полководца». – М.: Педагогика, 1990. – С.3-26.
128. Хаммер М., Чампи Дж. Реинжиниринг корпорации: манифест революции в бизнесе. Пер. с англ. - СПб: Изд. СПбГУ, 1997. - 332 с.
129. Хинчин А.Я. Педагогические статьи. - М.: Изд.АПН РСФСР, 1963. - 204 с.
130. Холодная М.А. Психология интеллекта. Парадоксы исследования. 2-е изд.перераб. и доп. - СПб.: Питер, 2002. - 272 с.
131. Хуторской А.В. Дидактическая эвристика. Теория и технология креативного обучения. - М.: Изд-во МГУ, 2003. - 416 с.
132. Хуторской А.В. Современная дидактика: Учеб. для вузов. – СПб: Питер, 2001. – 544 с.
133. Черепанов В.С. Экспертные оценки в педагогических исследованиях. - М.: Педагогика, 1989. - 152 с.
134. Шадриков В.Д. Деятельность и способности. - М.: Изд.корп. "Логос", 1994. - 320 с.
135. Шеннон Р.Ю. Имитационное моделирование систем - искусство и наука. - М.: Мир, 1978. - 419 с.

136. Ширяева В.А. Развитие системно-логического мышления учащихся в процессе изучения теории решения изобретательских задач (ТРИЗ). Автореферат Дис. ...канд.пед.наук. - Саратов: СГУ им. Н.Г.Чернышевского. 2000.
137. Шнейдерман Б. Психология программирования : Человеческие факторы в вычислительных и информационных системах. - М.: Радио и связь. 1984. - 304 с. (Оригинал: В. Shneiderman, Software Psychology: Human Factors in Computer and Information Systems. Cambridge: Winthrop, 1980.).
138. Юдина А.Г. Лого — среда для развития мышления // Информатика, № 33, 1995.
139. Якунин В.А. Обучение как процесс управления: психологические аспекты. - Л. 1988.
140. Янушкевич Ф. Технология обучения в системе высшего образования. - М. 1986.
141. Бабич И.Н. Информатика в начальной школе // Вопросы Интернет-образования. №3. ФИО. МЦИО. 2002.
URL: http://penza.fio.ru/vio/03/cd_site/Articles/art_5_7.htm
142. Бадд Т. Объектно-ориентированное программирование. Гл.1 Объектно-ориентированное мышление.
URL: <http://grizlyk.chat.ru/badd/index.htm>.
143. Битянова М.Р. Развитие проектного мышления.
URL: <http://psy.1september.ru/2002/30/3.htm>
144. Васютина Н.Ю. Компетентности и компетентностный подход в современном образовании
URL: <http://festival.1september.ru/articles/581708/>.
145. Городняя Л.В., Мурзин Ф.А. Психологические аспекты информатики. URL: <http://www.nsu.ru/archive/conf/nit/97/c5/node5.html>.
146. Давыдов В.В. Учебная деятельность: состояние и проблемы исследования.
URL: <http://www.voppsyl.ru/4y/ISSUES/1991/916/916005.php> .
147. Иванищина О.Н. Воображение и мышление у детей дошкольного возраста. На сайте Иванищина О.
URL: <http://ivalex.vistcom.ru/konsultac1.htm> Волгоград. 2003.
148. Ильясов А.Н. К вопросу о роли и месте программирования в школьной информатике.
URL: http://creonet.edu.ua/articles/art7_r.html .
149. Кашина Е.А. Курс лекций по дисциплине "Информатика". На сайте URL: http://www.useu-nt.ru/files/Klec_info.doc.

150. Кнут Д.Э. Алгоритмическое мышление и математическое мышление. URL: http://www.ihik.lib.ru/philosoph/ihik_1129.htm.
151. Копаев А.В. Алгоритм как модель алгоритмического процесса. Сайт проекта «Информика» URL: <http://www.visual.org.ua/articles.html> . 2003. 7 с.
152. Копаев А.В. Влияние современных информационных технологий на изучение основ алгоритмизации в средней школе. Там же. 10 с.
153. Копаев А.В. Модельная концепция обучения алгоритмике. Там же.
154. Копаев А.В. О практическом значении алгоритмического стиля мышления. Там же. 3 с.
155. Копаев А.В., Триус Ю.В. Фундаментальный аспект базового курса информатики. Там же. 3 с.
156. Лучко Л.Г. Поэтапное формирование умственных действий и знаний в обучении информатике.
URL: http://www.ompgu.omsk.edu/libdocs/it_sb2/igluchko.htm 3 С.
157. Рожина И.В. Психолого-педагогический анализ проблемы развития мышления учащихся при обучении объектно-ориентированному программированию.
URL:
http://bspu.ab.ru/Journal/vestnik/ARHIW/N1_2002/5_sekz/rojina.pdf
158. Степанова М.А. Проблема обучения и развития в трудах Л.С.Выготского и П.Я.Гальперина.
URL:
http://www.portalus.ru/modules/psychology/readme.php?subaction=showfull&id=1107780662&archive=1120045907&start_from=&ucat=27& .
159. Хазан О., Томэйко Дж. Рефлексия и абстракция в гуманитарных аспектах программирования // Открытые системы, 19.09.2005. URL: <http://www.osp.ru/os/2005/09/380361/>

Научное издание

Купчинаус Сергей Юрьевич

**ФОРМИРОВАНИЕ
КОНСТРУКТИВНО-ЛОГИЧЕСКОЙ КОМПЕТЕНТНОСТИ
БУДУЩИХ СПЕЦИАЛИСТОВ ИНФОРМАТИКИ И
УПРАВЛЕНИЯ**

Монография

Авторская редакция

Подписано в печать 30.11.11 Формат 60х84 ¹/₁₆

Печать офсетная. Усл.печ.л. 8,84

Тираж 30 экз. Заказ №

Издательство «Удмуртский университет»
426034, г. Ижевск, ул. Университетская, 1, корп. 4.
Тел./факс: +7(3412) 500-295 e-mail: editorial@udsu.ru